TAMPERE UNIVERSITY OF TECHNOLOGY

LUIS ENRIQUE GONZALEZ MOCTEZUMA

CYBER SECURITY ASSESSMENT FOR WEB SERVICE-BASED MONITORING OF INDUSTRIAL SYSTEMS

Master of Science Thesis

Examiner: Professor José L. Martínez Lastra
Examiner and topic approved in the
Automation, Mechanical and Materials
Engineering Faculty Council meeting on
05 May 2010

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master of Science Degree Programme in Machine Automation
GONZALEZ MOCTEZUMA, LUIS ENRIQUE: Cyber Security Assessment for Web Service-based Monitoring of Industrial Systems
Master of Science Thesis, 99 pages, 4 Appendix pages
February 2011
Major: Factory Automation
Examiner: Professor José L. Martínez Lastra
Keywords: industrial cyber security, industrial monitoring, Web Services, DPWS security, security framework, WS Security suite, factory automation

The adoption of Web Services in industrial systems promises a new level of reconfigurability, scalability and interoperability among industrial devices and applications. However, it also opens the door to many of the threats and cyber attacks known in the IT world. The aim of this thesis is to identify the mechanisms, techniques and knowledge, existing in traditional IT cyber security, which can be applied to improve the cyber security of an industrial system. This work considers strictly the fact that the Web Services are deployed at device level, thus the selection of security mechanisms and architectures has to fit with the resource constrained nature of this application.

It is important to understand the system that needs to be protected; therefore this work extracts the main characteristics and requirements of industrial systems deployed under centralized and distributed layouts. Security mechanisms used in IT cannot be applied seamlessly at device level; thus a feasibility study is done around implementing security in DPWS devices and different security mechanisms are assessed and chosen so that they adequate to the constraints imposed by the thesis application.

Without doubt, one of the most outstanding properties of the DPWS stack is its alignment and interoperability with Web Service specifications. This thesis exploits this benefit by describing how different protocols from the Web Services Security suite can be used to provide a variety of security services on the target system.

This work explains how lot of the threats known in the IT world are applicable to an industrial system that uses Web Services for monitoring. However, an important contribution of the thesis is to point the vulnerabilities that arise when Web Services are used at device level in an industrial environment.

It is not possible to provide a security framework that covers all possible threats for all types of industrial systems. The security framework has to be tuned and configured depending on the application needs. The final contribution of this work is a set of guidelines, in the form of decision diagrams, which help designers to select the architectural components and security protocols depending on the requirements and characteristics of a given system.

PREFACE

Not always it is possible to do what you like the most; that is why I want to thank to all those 1000 persons who listened 1001 complaints about this document writing. Despite of that I did a big effort to glue, in a congruent and formal manner, the topics I was entrusted.

I appreciate the knowledge and support I got from my colleges at the FAST Lab, whom helped me to improve my engineering skills. I want to give special thanks to my thesis supervisor Jani Jokinen for his advices and feedback on this work.

My deepest gratitude goes to Professor Lastra. I want to thank him for giving me the opportunity to collaborate in his working group, make possible this master programme and being a cornerstone in my academic life.

An special acknowledge to all those persons and organizations that made possible the SAMIA project, which was used as reference to design and develop this thesis.

I want to thank a bunch of special persons (almost 1000), I hope I do not miss any: my beloved lady aka Martita, Marrano, Mr., Gabito, el Chino, Palo, Gagari, Aki, el Buki, Anita, Piotr, Aija, Andre, Padrino, JC, Popon, Popita, Angelica, Axel, Lauchini, Nuria, Laurita, Jess, Vaiva, Bern, Bety and Frodo.

Finally I want to thank my family, each of you (Brody, Mama, Lalo, Itzelita, Melita and Pollo) for always answering the phone when here the sun sets and there just rises.

Luis Enrique Gonzalez Moctezuma
Riga, February 2011

CONTENTS

1 INTRODUCTION 1
1.1 Background 1
1.2 Motivation 3
1.3 Problem Definition 4
1.4 Work Description 4
1.4.1 Thesis Domain 4
1.4.2 Thesis Goals 5
1.4.3 Methodology Description 5
1.4.4 Scope and Assumptions 6
1.5 Thesis Organization 7
2 THEORETICAL BACKGROUND 8
2.1 Web Services 8
2.1.1 Core Technologies 9
2.1.2 Web Services Architecture 11
2.1.3 DPWS 14
2.1.4 Web Services in Industrial Systems 15
2.2 Security 18
2.2.1 Security Fundamentals 18
2.2.2 Security Mechanisms 20
2.2.3 Tunneling Protocols and Firewalls 23
2.2.4 Web Services Security 25
2.2.5 DPWS Security 30
2.2.6 Information Security in Industrial Systems 34
2.3 3G Mobile Communication 39
2.3.1 Security 40
3 SCENARIOS DESCRIPTION 42
3.1 Framework 43
3.2 Centralized Layout 45
3.3 Distributed Layout 46
3.4 Scenario A 47
3.5 Scenario B 49

3.6 Monitoring Service with Multiple Intermediaries 50
4 SECURITY ASSESSMENT 52
4.1 System Requirements 52
4.1.1 Functional Requirements 52
4.1.2 Security Requirements 54
4.2 Feasibility to Implement Security in DPWS Devices 59
4.2.1 DPWS Devices Constraints and Capabilities 59
4.2.2 Security Scope and Extensibility of the DPWS Stack 62
4.3 Security Mechanisms Assessment 63
4.3.1 Miscellaneous Mechanisms 63
4.3.2 Web Services Security Suite 67
4.4 Threats Identification 73
4.4.1 Spoofing Attack 73
4.4.2 Eavesdropping Attack 75
4.4.3 Logon Abuse Attack 75
4.4.4 DoS Attack 75
4.4.5 Application Level Attack 77
4.4.6 Radio Jamming 77
4.5 Secure Framework for WS-based Monitoring of Industrial Facilities 78
4.5.1 Architectural Components 79
4.5.2 Security Protocols 83
5 FUTURE WORK 87
5.1 Implementation of Security in DPWS Devices 87
5.2 Security Code Generator for DPWS Targets 87
5.3 Performance Analysis in an Industrial System 88
5.4 Simulation of Secure Transactions in an Industrial System 89
5.5 Attack Signatures for Industrial Systems 90
6 CONCLUSIONS 91
7 REFERENCES 94
8 APPENDIX A - WS standards overview 100
9 APPENDIX B – Interaction of WS specifications 103

ABBREVIATIONS

BPEL	Business Process Execution Language
DMZ	Demilitarized Zone
DoS	Denial of Service
DPWS	Devices Profile for Web Services
ECC	Elliptic Curve Cryptography
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IDS	Intrusion Detection System
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Network
OASIS	Organization for the Advancement of Structured Information Standards
OEE	Overall Equipment Effectiveness
OPC UA	OLE for Process Control Unified Architecture
OSI	Open Systems Interconnection
PKI	Public Key Infrastructure
SAMIA	Service-bAsed Monitoring for Industrial Ambients
SAML	Security Assertion Markup Language
SCADA	Supervisory Control and Data Acquisition
SIRENA	Services Infrastructure for Real-time Embedded Networked Applications
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol

SOCRADES	Service-Oriented Cross-layer infRAstructure for Distributed smart Embedded devices
SODA	Service Oriented Device and Delivery Architecture
TCP	Transmission Control Protocol
TLS/SSL	Transport Layer Security / Secure Sockets Layer
UDDI	Universal Description, Discovery and Integration
VPN	Virtual Private Network
WAP	Wireless Application Protocol
WS	Web Service
WS-I	Web Services Interoperability Organization
WS-*	Web Services set of specifications
WSDL	Web Service Description Language
W3C	World Wide Web Consortium
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
3G	3rd Generation of mobile telecommunications

LIST OF FIGURES

Figure 1.1. Thesis domain. 4
Figure 2.1. Communication of IT components through WS 8
Figure 2.2. Example of SOAP message on top of HTTP. 10
Figure 2.3. Interaction of WS core technologies. 11
Figure 2.4. WS Standards Stack 12
Figure 2.5. WS message composeability 13
Figure 2.6. Device and hosted services in DPWS specification 14
Figure 2.7. Protocol stack of DPWS. 15
Figure 2.8. WS at device level in the shop-floor. 17
Figure 2.9. Confidentiality in asymmetric encryption. Modified from 22
Figure 2.10. VPN elements and operation. Modified from 24
Figure 2.11. Typical firewall arrangement 25
Figure 2.12. WS Security Standards. 26
Figure 2.13. XKMS topology 27
Figure 2.14. General WS Security messaging 30
Figure 2.15. DPWS Standard security model 31
Figure 2.16. Architecture on the SODA security framework 32
Figure 2.17. DPWS-based device with security services support 32
Figure 2.18. Secure orchestration of WS with APEL. 33
Figure 2.19. Typical security mechanisms in an industrial system 34
Figure 2.20. ISA 99.00.02 Cyber Security Management System.. 36
Figure 2.21. Defense-in-depth for a SCADA system 38
Figure 2.22. Simplified 3G architecture 40
Figure 2.23. 3G security architecture 41
Figure 3.1. Monitoring of industrial ambients with web services. 44
Figure 3.2. Scenario A, manufacturing system. 47
Figure 3.3. Network arrangement of Scenario A. Modified from. 48
Figure 3.4. Monitoring unit used in scenario B.. 49
Figure 3.5. User interface used in Scenario B. 50
Figure 3.6. Industrial monitoring with multiple intermediaries. 51
Figure 4.1. Subscribing to DPWS controllers from external networks. 54

Figure 4.2. Point-to-point and end-to-end security 57
Figure 4.3. DPWS-based controller with security mechanisms. 60
Figure 4.4. SSL performance in resouce-constrained devices 61
Figure 4.5. WS Security suite within the DPWS stack of an industrial controller. 68
Figure 4.6. WS-Trust in a monitoring application of industrial components. 70
Figure 4.7. WS-SecureConversation in the monitoring of an industrial component. 72
Figure 4.8. Security vulnerability by using discovery in DPWS devices. 74
Figure 4.9. DPWS industrial unit under DoS attack. 76
Figure 4.10. Decission diagram for choosing the architectural components of the secure framework. 81
Figure 4.11. Secure components with Defense-in-depth arrangement for a centralized monitoring case. 82
Figure 4.12. Secure components with Defense-in-depth arrangement for a distributed monitoring case. 83
Figure 4.13. Decission diagram for choosing the security protocols of the secure framework. 85
Figure 5.1. Security code generator for DPWS targets. 88

LIST OF TABLES

Table 2.1. Relationship of Security Services and Layers 1-7 19
Table 2.2. Latency measurements for encryption algorithms in an embedded industrial controller. 39
Table 3.1. Event-based Vs Polling-based monitoring..... 43
Table 3.2. Comparisson between centralized and distributed monitoring layouts. 47
Table 4.1. Monitoring messages type, priorities and tolerated delays. 53
Table 4.2. Security services relevance 56
Table 4.3. Threats in the monitoring of industrial environments with WSs. 78

1 INTRODUCTION

This chapter presents the background, the motivation for realizing this thesis, a description of the work and the organization of this document.

1.1 Background

In recent years, industrial automation has benefited from the improvements achieved in embedded systems. Thanks to the miniaturization of electronic components, nowadays it is possible to have industrial devices with smart capabilities and a broad range of communication interfaces. Having such computational resources at device level allows to shift industrial functions from a centralized schema to a distributed one. For example, valves with autodiagnosis functionalities can notify about their maintenance needs or components within an industrial facility can be controlled individually by using embedded networked controllers which can operate in a coordinated manner. All this is evolving the way industrial automation systems are designed, maintained and operated.

For years industrial vendors have adopted a business model based on proprietary systems. This is a big obstacle if industrial components are required to interact between them (horizontal integration) and communicate with high level applications (vertical integration) like Manufacturing Execution Systems (MES) or Enterprise Resource Planning (ERP) systems. In order to allow interoperability across devices and platforms, an open communication technology is needed. Web Services (WSs) at device level is forecasted as a serious candidate to facilitate this.

The Devices Profile for Web Services (DPWS) is a communication protocol stack which provides WSs communication capabilities on resource-constrained devices. One of its promises is to inherit the benefits of using WSs like interoperability, flexibility, scalability, reconfigurability, composition and security. Another advantage is that lot of the development tools, frameworks and knowledge acquired in WSs for the Information Technology (IT) field can be applied to WSs for industrial systems. For instance DPWS allows to apply the Service Oriented Architecture (SOA) design paradigm in industrial systems by using WSs as the core technology for communications (Jammes, Mensch and Smit 2007).

Lately, lots of research efforts have been focused on demonstrating the feasibility and benefits of implementing SOA at the factory floor by using DPWS-enabled industrial devices. Projects like *Services Infrastructure for Real-time Embedded Networked* (SIRENA) demonstrated the coordination and control of an industrial demonstrator where its components use the DPWS, allowing its functionalities to be exposed and invoked through WSs (Bohn, Bobek and Frank 2006). The *Service-Oriented Cross-*

layer infRAstructure for Distributed smart Embedded devices (SOCRADES) project demonstrated how communication gaps between the factory level and business level can be minimized when industrial devices with WS interfaces are used (Cannata, Gerosa and Taisch 2008).

Traditionally monitoring in Supervisory Control and Data Acquisition (SCADA) systems is done by polling the sensors readings or equipments measurements. Nevertheless, a device that implements the DPWS has the possibility to communicate through events. The data contained in those events can be consumed and processed by any application or system in a standard form. This allows to have event-based monitoring with rich data coming directly from the industrial devices and sensors (SAMIA 2009).

An application that uses WSs for monitoring, takes the advantage that WSs are transported over Hypertext Transfer Protocol (HTTP). This protocol is widely used and empowers a huge amount of network transactions across the Internet. Messages transported over HTTP can cross easily firewalls as outgoing traffic. This empowers industrial monitoring across a low cost network, like Internet, without major modifications (if any) to the system architecture.

Industrial monitoring encompasses the utility sector, which usually is characterized by having installations in remote places which are physically difficult to access. Monitoring with wired technologies is unfeasible for this kind of applications; wireless communication is required. A good candidate is 3G networks. The big momentum of mobile telecommunications has forced service providers to create robust wireless infrastructure with extensive coverage area. 3G networks can provide wireless voice and data services. Data services enable a mobile endpoint to communicate through Internet. 3G supports communications which are built on top of the Internet Protocol (IP); this includes HTTP and therefore WSs. An industrial device that implements the DPWS can communicate the monitoring events if it is interfaced through a 3G modem.

Industrial systems are critical infrastructures because they involve public services, production systems, machinery, processes and human labor. As critical infrastructures, the compromise of any of its assets (physical or logical) can risk the integrity of personnel or the system itself. Threats can occur at different points and through different vectors. In order to prevent threats occasioned by security holes in the information system it is necessary to apply cyber security mechanisms.

In the past, industrial systems were not connected between them, neither connected to public networks like Internet; this isolation provided a protection layer against outsider attacks. Security designers and vendors trusted in *security by obscurity* which states that the lack of detailed technical information and knowledge for an attacker can prevent it from exploiting the protocols and systems vulnerabilities (Dzung, et al. 2005).

Nowadays, the most common schema to built cyber security around industrial systems is by using *defense-in-depth* security. In this schema security layers or shells are built around the assets that need to be protected. Industrial security services are oriented on data integrity and they implement simple authentication and authorization mechanisms which are desired in control applications.

Nevertheless, the evolution of industrial systems by introducing WSs at device level raises new security threats, constraints but also possibilities. The knowledge acquired on the IT field can be applied. The security built around WSs should be exploited and applied at device level.

1.2 Motivation

Most of the research efforts have been concentrated in making functional and practical the implementation of WSs at device level. Others have exploited the reconfigurability, scalability features that are inherited when using this technology at the shop-floor. But very little research has been conducted in security for industrial devices that implement the DPWS. For instance not physical implementation of DPWS with security services has been done yet.

Industrial devices that implement the DPWS will not be able to reach an operational stage in commercial applications if security services are not integrated into them. As already reviewed, an industrial environment encompasses critical infrastructure, thus the protection of its physical and information assets is a high priority issue.

Interconnection of applications and devices is a major driver in modern industrial systems. The adoption of open standards, like WSs, facilitate the required interoperability. Therefore the security assumptions (isolation and obscurity) that were used for previous industrial environments are not valid anymore.

If it is true that lot of the frameworks, tools and knowledge acquired in WSs for traditional IT applications can be applied in WSs at device level, it is also true that industrial applications have a number of distinct security-related requirements compared with traditional IT systems.

It is well known that security mechanisms are computational demanding tasks. The resource-constrained nature of DPWS devices imposes new limitations to the application of well proved and tested security mechanisms/frameworks by the IT community. Therefore, it is necessary a selection of selection of mechanisms/frameworks that adequate to the requirements and constraints of an application that employs DPWS devices.

Even though, DPWS devices introduce new constraints, they also offer interesting potential solutions to deal with cyber security issues. For instance, the DPWS stack should be exploited and leverage the security mechanisms that are offered by the WSs security suite. An analysis of how security on WSs can be used for the requirements of industrial monitoring is needed.

It is important to adapt and implement security to the new needs and constraints of modern industrial systems. It is not possible to use previous assumptions used in security for old industrial systems. Neither is feasible to apply seamlessly the security methods used in traditional IT systems.

1.3 Problem Definition

Analyze how the introduction of an emerging technology, like WSs at device level, impacts the cyber security of industrial monitoring systems. The requirements and threats that arise should be identified. It is also needed to select a set of secure mechanisms and frameworks that adequate to the application needs.

1.4 Work Description

This section defines the study domain of this thesis and its relations with other IT areas. A description of the thesis goals and the used methodology is given. The scope and assumptions are detailed.

1.4.1 Thesis Domain

This thesis belongs to the domain of industrial automation and it is focused on monitoring applications. WSs are used as the underlying communication technology that enables the monitoring process. In order to have WS capabilities in industrial systems, devices and controllers that implement the DPWS are used. This thesis analyzes and discusses security in WS-based monitoring of industrial systems.

In order to achieve this, three major areas are merged: The knowledge and techniques used in cyber security for traditional IT applications, the architectures and mechanisms used in industrial cyber security and the area that deals with the implementation of WSs at device level. All these three areas are combined and analyzed under the context and requirements of industrial monitoring. The diagram illustrated in Figure 1.1, shows the three core areas of this thesis, the intersection of them is the target to be covered in this thesis. The dotted box defines the context under which the analyzes and assessments are done; industrial monitoring.

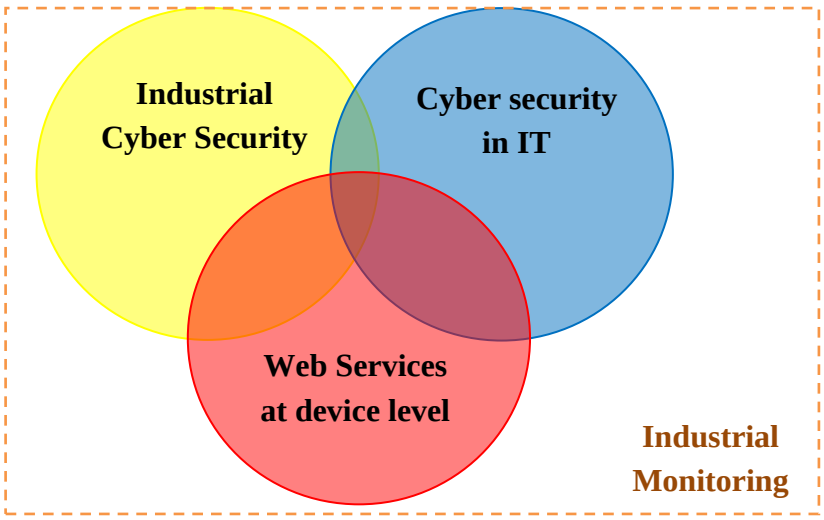


Figure 1.1. *Thesis domain.*

1.4.2 Thesis Goals

The thesis goals are:

1. Characterize industrial systems which can be found under centralized and distributed layouts.
2. Define the functional/security requirements for the monitoring of industrial environments by using WSs.
3. Analyze the feasibility to implement security services in WSs at device level.
4. Review and evaluate different security mechanisms that can be used to provide security services in an application that employs WSs for monitoring an industrial environment.
5. Identify the cyber security threats that arise when an industrial environment is monitored with WSs at device level.
6. Propose guidelines to configure a secure framework (components and protocols) depending on the characteristics/requirements of the monitored system.

1.4.3 Methodology Description

In order to accomplish the thesis goals, the next methodology was followed:

Literature review of Cyber Security

A coarse review of these topics was done in the IT field. Security services, attacks and countermeasure mechanisms are studied and gathered. Technologies for securing WSs are reviewed. Then the state of art in security mechanisms in devices that implement DPWS is surveyed. Finally a review of security architectures and mechanisms for protecting the cyber security of industrial facilities is done.

Scenarios description

This thesis uses as reference two real implementations of industrial environments monitored with WSs. One of the monitored scenarios is a discrete manufacturing line while the other is a water tower. Each of the scenarios is representative of layouts that can be found commonly in industrial systems: centralized and distributed layouts. For instance process and manufacturing industries are example of centralized systems while the facilities used in the utility sector are arranged in a distributed layout.

The general characteristics of industrial systems under centralized and distributed layouts are derived out of the scenarios.

A monitoring application that involves multiple intermediaries is proposed in order to highlight the need to use end-to-end security in order to provide security services and still ensure the reconfigurability and scalability of the system.

Requirements and feasibility of implementing security in DPWS devices

The characterization generated in the previous point is used as starting point to identify the functional and security requirements for applications that monitor industrial environments under centralized and distributed layouts with WSs.

The feasibility to implement security services in industrial controllers that implement the DPWS is analyzed and the scope and extensibility, of the DPWS specification, in security terms is discussed.

Security mechanisms assessment and threats identification

Considering the requirements and feasibility analyzed in the previous point, two filters are applied to the security mechanisms gathered during the literature review process. Firstly, those with characteristics that make them suitable to be employed for securing a system that employs DPWS devices are chosen. Secondly, on those who passed the first filter, a discussion is done on how they can be used for the monitoring application. Their advantages, disadvantages and feasibility to be deployed on centralized and distributed layouts are analyzed.

Threats that arise when WSs are used for monitoring an industrial system are identified. Traditional IT threats are visualized for an industrial communication system. Other threats proper of an application that employs DPWS devices are identified.

Secure Framework proposal

It is not possible neither practical, to offer an unique secure framework that will protect every monitored system from all the identified threats. The secure framework has to be tuned and adapted depending on the characteristics of the monitored system. To do this decision diagrams are proposed in order to facilitate the selection of the architectural components and the security protocols based on the industrial system characteristics/requirements.

Future work

After reviewing sources and documentation for this thesis, the opportunity areas or uncovered gaps are identified and proposed as future for work for further research projects.

1.4.4 Scope and Assumptions

1. The term security in this document refers to the security in the IT domain. In literature this is also referred as cyber security.

2. Different security mechanisms are analyzed, but a special interest and importance in the analyzes is given to those security mechanisms built on top of WSs, since the thesis novelty is to use this technology for monitoring purposes.

3. The monitoring of the industrial systems is done under the event-based schema, where the monitored data is pushed out of the system every time an event or specific condition is met within the system. Therefore all the monitoring messages are originated inside the industrial facilities. In this way, polling-based monitoring is out of the scope of this thesis.

4. This thesis does not intend to implement security protocols and mechanisms in a physical system.

1.5 Thesis Organization

The rest of the document is organized as follows: Chapter 2 introduces the most relevant concepts and technologies used in this work. Chapter 3 introduces the scenarios that are used as reference for the characterization of centralized/distributed industrial systems. In Chapter 4 develops a security assessment and analysis for industrial systems which monitored with WSs. Chapter 5 proposes future work that can be developed in the field of security in embedded devices used in industrial systems. Finally the conclusions of this thesis are presented in Chapter 6.

2 THEORETICAL BACKGROUND

This chapter is divided in three sections: Web services, security and 3G networks. The first section includes an overview of the core communication technology of this thesis, Web Services. It also describes the Device Profile for Web Service which enables the implementation of WSs in electronic devices like industrial controllers and finalizes with review of the use of WSs in industrial environments. The second section (security), is conformed by security in distributed systems, security in Web Services, the state of the art in security for DPWS devices and cyber security for industrial systems. The last section briefly describes 3G mobile networks, a review of security on them is also presented. This is due to the fact that one of the case scenarios uses 3G networks to transport the monitored data thus, this information is needed for a later analysis.

2.1 Web Services

WSs can be defined as self-contained, modular applications which can be described, located, published and invocated over a network (Cui, et al. 2009) in order to execute an encapsulated function. Their primary objective is to allow interoperability and integration between distributed, heterogeneous and autonomous IT components. WSs enable communication across platforms, operating systems, and programming languages

With the previous in mind, reusability of software is another big advantage, since the functionalities of legacy applications (Benatallah, et al. 2005) can be exposed through WSs, allowing other components (applications, devices) to use them locally or remotely, for example through internet, as shown in Figure 2.1 where a Web Server exposes some of its functionalities as WSs.

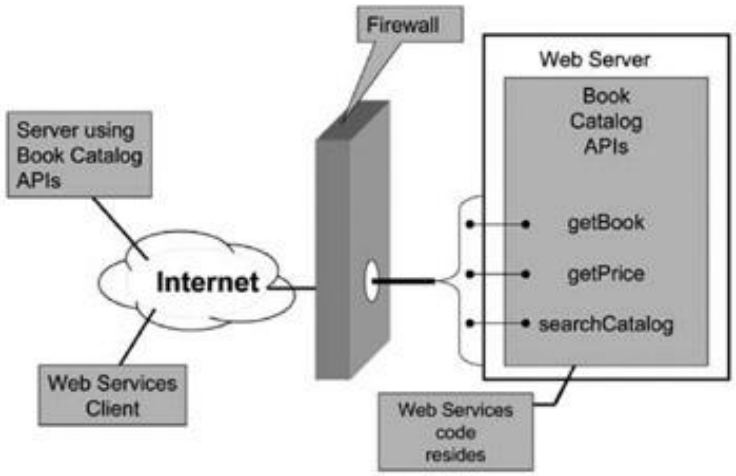


Figure 2.1. *Communication of IT components through WSs (Shah 2007).*

The most common implementation of WSs is done with Simple Object Access Protocol (SOAP), an eXtensible Markup Language (XML) based protocol which is transported over HTTP (Shah 2007); thanks to this WS messages can be exchanged seamless within a Local Area Network (LAN) or they can be routed through internet (Alonso 2004). This is what makes out of WSs a very flexible technology to enable horizontal and vertical integration.

Once WSs are deployed, they can be aggregated in what is called a composite of services (Zeng, et al. 2003). This is done when a high level service depends on the functionalities provided by other services. The classic literature example is the travel planner service, which can be a composite of multiple component services like: flight booking, travel insurance and car rental. These services can be distributed among different providers, and encapsulated in just one service, the travel planner service. QoS plays an important roll when services are compounded, since the overall QoS of the composite service depends on the QoS of each of the encapsulated services.

Another key point of WS is that they use open standards generated by bodies like the World Wide Web Consortium (W3C), Organization for the Advancement of Structured Information Standards (OASIS), or Internet Engineering Task Force (IETF), among others. This allows a coordinated and continuous development of standards to address new necessities (Cui, et al. 2009) without falling in the pitfalls of using proprietary communication protocols. This is what has motivated major software vendors to move their applications interfaces to WSs (Newcomer and Lomow 2007).

The technologies that enable WSs are described next. Then the WS standards set is introduced; this topic has special relevance for this thesis since WSs have a rich set of standards that can be used to provide security characteristics on the application of this thesis. Finally the use of WSs in the industrial domain is reviewed.

2.1.1 Core Technologies

The technologies that allow the implementation of WSs are introduced next.

SOAP

SOAP is a W3C standard that provides the definitions to exchange structured and typed information between peers in decentralized and distributed environments (W3C 2007). It is XML-based, which makes it human readable, machine understandable, simple and extensible. For example XML techniques can be applied to SOAP messages, like XML encryption (that increases the security of the communication).

A SOAP message is an XML document which contains the next elements:

- A SOAP Envelope element, where the XML document is identified as a SOAP message
- A SOAP Header element where application-specific information is placed

- A required SOAP Body that contains the SOAP message that will be delivered to the last endpoint.
- A SOAP Fault element to notify errors.

As it was mentioned before, SOAP messages are usually transported on top of HTTP. Figure 2.2 shows and example of a simple SOAP message, the lines in red color correspond to the transport protocol HTTP and the blue lines to the SOAP message, where the operation *GetTemperature* is requested. It should be considered that SOAP messages transported with HTTP inherit the security weakness, bugs and holes of the HTTP protocol. On the other hand, SOAP messages can pass through firewalls facilitating the communication process.

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body xmlns:m="http://www.example.org/weather">
  <m:GetTemperature>
    <m:City>Tampere</m:City>
  </m:GetStockPrice>
</soap:Body>
</soap:Envelope>
```

Figure 2.2. *Example of SOAP message on top of HTTP.*

Web Service Description Language (WSDL)

WSDL is a W3C standard, used to describe the capabilities of a WS (Shah 2007). It is used to list the operations that a WS can provide and the notifications to which a client can subscribe. The structure of the input and output messages, used to invoke operations and receive notifications, are expressed with this standard. It is also used to indicate the location of the WS on the network.

For practical implementations all this information is written in a WSDL file. This file is used by the service client to locate, invoke and subscribe properly to the WS.

Universal Description, Discovery and Integration (UDDI)

UDDI is a directory that stores information about a set of WSs. This information is described by the WSDL of the WS. The communication with this repository is via SOAP. UDDI are especially relevant when a service client looks for a WS without prior know-

ledge of where the service is and how to interact with it. This process goes as follow (Shah 2007):

- First the service client, request for services that meet a set of requirements to the UDDI.
- Then the UDDI responds with the location of where to find the WSDL of the available WSs.
- The service client chooses a WS provider, looks for its WSDL file in order to know how to interact with the WS.

Figure 4 illustrates the three core technologies that empower WSs and the relations between them.

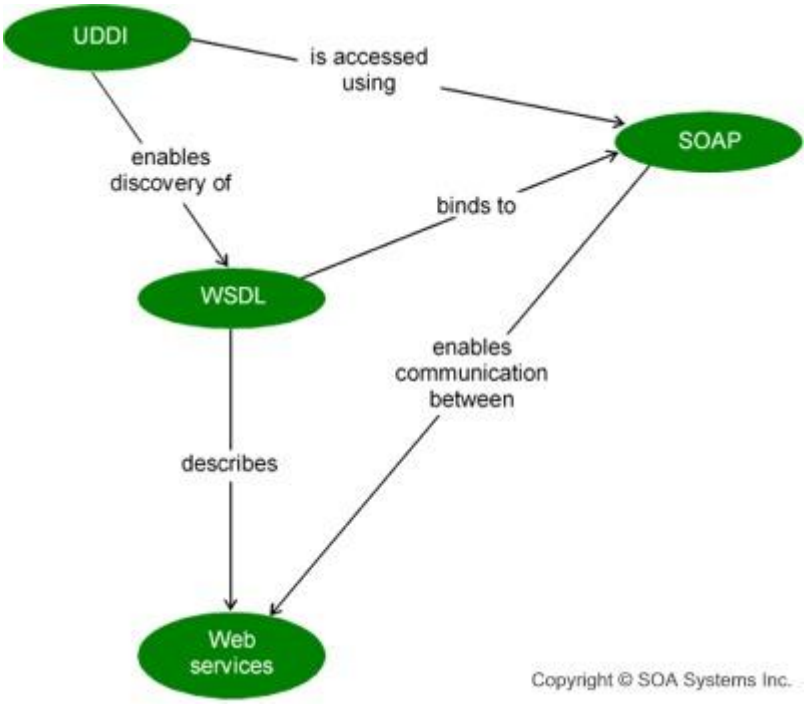


Figure 2.3. *Interaction of WSs core technologies (SOASPECS 2009).*

2.1.2 Web Services Architecture

Due to the big adoption and use of WSs, the list of demands and requirements in specific areas grew. It was necessary to address specific issues like security, transactions or reliability. This is why WS specifications and standards are created, to cover special needs. WS specifications are usually started by companies or software vendors which join together to write a specification. Then the specification is submitted to a standardization body like W3C, OASIS or IETF with the aim to make out of the specification a standard. Microsoft and IBM are the de facto leaders of WS specifications (Newcomer and Lomow 2007).

The WSs architecture is composed by different WS standards that can be used to address different needs. These standards are grouped in categories as shown in Figure 2.4. Within each of these categories, there is a set of standards/specifications. The set of all these standards is usually referred as the WSw Standards stack, usually abbreviated as WS-*. It is important to mention that the primary specifications that should be included in an implementation to allow WS communications are those grouped in the Transport, Invocation and Description categories. Additional specifications can run on top of these ones. This ability to compose WS specifications will be explained in the next topic.

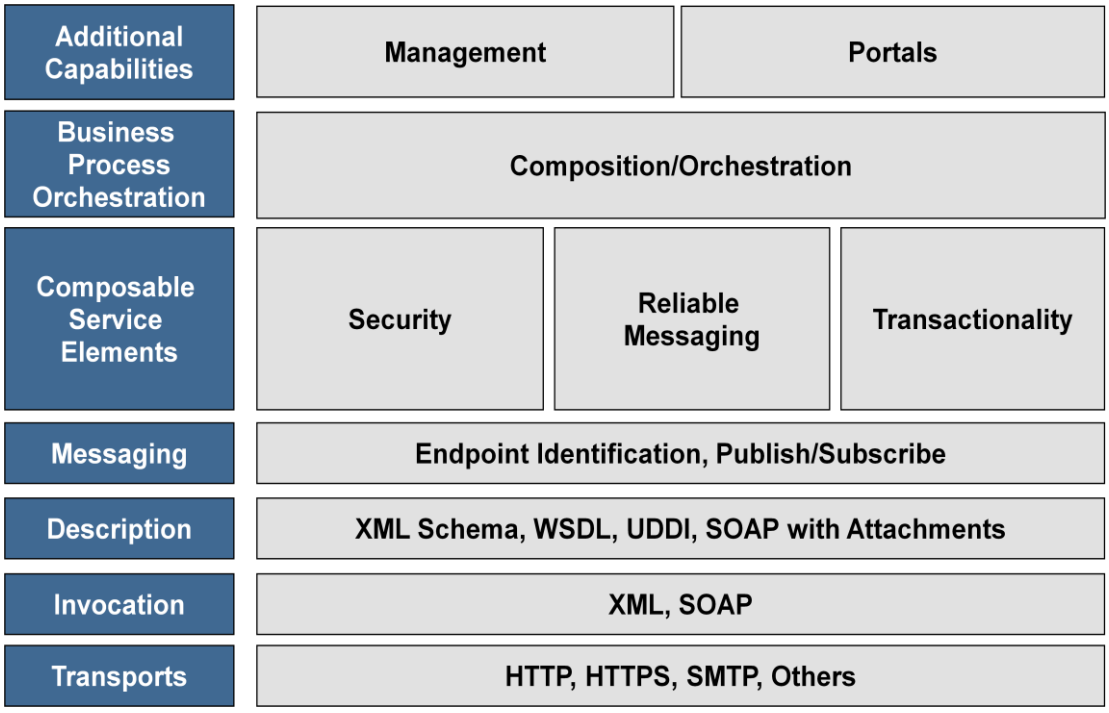


Figure 2.4. *WSs Standards Stack (WS-I 2007).*

APPENDIX A - WS standards overview, includes a more detailed list of standards which are relevant for this thesis, especially those related with security and QoS issues, despite that they will be covered in the Security section of this chapter.

Specification composeability

One of the cornerstones of WSs is that their specifications are composeable (which should not be confused with the composition of services, previously reviewed), meaning that independent WSs specifications can be combined to provide an implementation that addresses a broader set of requirements. In this way developers can extend their implementations depending on the evolution of their application needs.

One important characteristic, achieved by using XML-based documents, is the multipart message structure. This kind of structure allows an easily implementation of composition since adding new message elements does not affect the overall message functionality. For instance, a communication protocol that implements transactions protocols can be extended with reliability and vice versa, without having protocol conflicts.

As an important remark, WSDL and SOAP, two basic WS specifications, can support composition natively (Weerawarana, et al. 2005).

Figure 2.5 shows a multipart WS message (SOAP message), the example is composed by three WS specifications: WS-Adressing, WS-Security and WS-ReliableMessaging. Each element of the message is independent from the others and they address issues like delivery of messages, security and reliability.

```
1  <S:Envelope...>
2    <S:Header>
3      <wsa:ReplyTo>                                                    } WS-Addressing
4        <wsa:Address>http://business456.com/User12</wsa:Address>
5      </wsa:ReplyTo>
6      <wsa:TO>HTTP://Fabrikam123.com/Traffic</wsa:To>
7      <wsa:Action>http://Fabrikam123.com/Traffic/Status</wsa:Action>
8      <wssec:security>
9        <wssec:BinarySecurityToken                                     }
10           ValueType="wssec:x509v3"                                   }
11           EncodingType="wssec:Base64Binary"                          } WS-Security
12           dXJcY3TnYHB....Ujmi8eMTaW                                  }
13       </wssec:BinarySecurityToken                                    }
14     </wssec:Security
15       <wsrm:Sequence>                                                }
16           <wsu:Identifier>http://Fabrikam123.com/seq1234</wsu:Identifier> } WS-Reliable Messaging
17           <wsrm:MessageNumber>10</wsrm:MessageNumber>                }
18       </wsrm:Sequence>
19 </S:Header>
20 </S:Body
 •   <app:TrafficStatus
 •     xmlns:env="http//highwaymon.org/payloads">
23           <road>520W</road>
24           <speed>3mph</speed>
25   </app:TrafficStatus>
26 </S:Body>
27 </S:Envelope>
```

Figure 2.5. *WS message composeability (Weerawarana, et al. 2005).*

APPENDIX B – Interaction of WS specifications shows an interaction diagram that describes how different WS specifications are related and how they can be composed in an implementation that uses WS.

Profiles

As already mentioned, the use of WSs is an emergent market, various companies and standards organizations (W3C, OASIS or IETF) are continuously working with the aim of publishing WS specifications or standards, thus issues related to ambiguity of interpretation, functionality overlapping and insufficient understanding of interactions between them (Weerawarana, et al. 2005) are obstacles for allowing interoperability when WS specifications are composed.

In order to cope with interoperability in the scope of WS specifications, profiles are defined. A profile includes a stack of WS specifications, recommended guidelines of how to use them together and the exclusion of inadequate specified features (Weerawarana, et al. 2005). Profiles are useful instruments to achieve interoperability, they simplify the discussion of WS at different granularity levels and clarify ambiguities.

The biggest player in WS interoperability is the Web Services Interoperability Organization (WS-I), an open industry organization, which has created the WS-I Basic Profile (WS-I 2010). Other profiles that enhance areas like security are built on basis of this profile like the WS-I Reliable Secure Profile.

The profile that is used in this thesis is the Device Profile for Web Services which is optimized to integrate WS in resource-constrained devices and it is explained next.

2.1.3 DPWS

DPWS is a device-targeted WS protocol stack profile, that was started by Microsoft and since June 2009 became a standard by OASIS. DPWS enables WS capabilities on resource-constrained devices. Such capabilities include, secure invocation of WS operations, description and dynamic discovery of WS and mechanisms to subscribe and receive events from WS. It should be highlighted that devices that implement the DPWS are fully aligned with the WS technology. At the time of this writing DPWS 1.1 is the standard supported by OASIS (OASIS 2009).

The DPWS specification defines two main elements: the device and its hosted services. The devices are important for the discovery and metadata exchange protocols. The services, which are hosted in the device, encapsulate the functionalities that the device can provide (Candido, et al. 2010). These elements are represented in Figure 2.6.

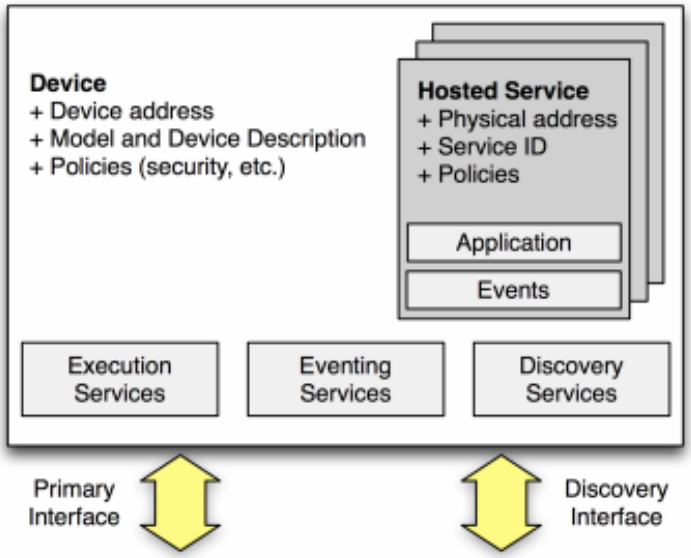


Figure 2.6. *Device and hosted services in DPWS specification (Candido, et al. 2010).*

DPWS is composed by WS specifications in a similar structure to the WS architecture, previously reviewed. Figure 2.7 shows the stack of protocols that composes the DPWS. Due to them it is possible to provide the next services (OASIS 2009):

- ***Discovery***: by using WS-Discovery, a device can advertise itself and discover other devices in the network.

- ***Messages addressing***: WS-Addressing provides a mechanism to asynchronously exchange SOAP messages among endpoints.

- ***Metadata exchange***: with WS-MetadataExchange it is possible to host services descriptions that can be accessed in form of WSDL and XML schemas.
- ***Policies description***: WS-Policy facilitates the description of the service capabilities, requirements and characteristics.
- ***Event publish/subscribe***: by using WS-Eventing, devices can subscribe to asynchronous messages which are produced when an event occur.
- ***Security***: although WS-Security is not formally part of the profile, it is the most accepted specification to implement security. DPWS security can also be extended by using WS-Trust and WS-SecureConversation.

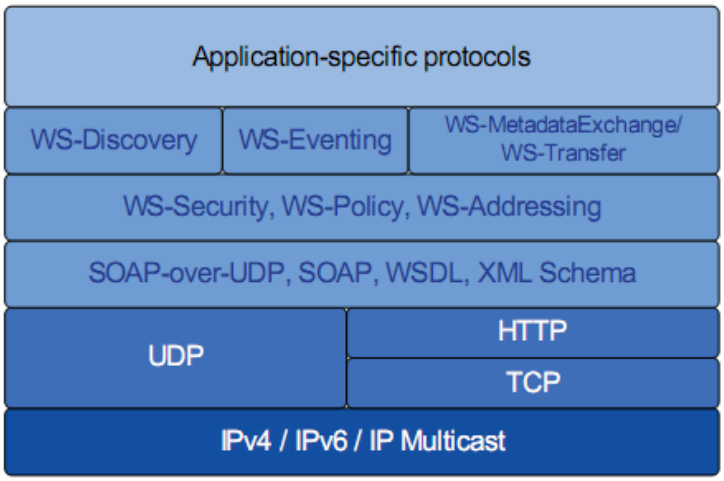


Figure 2.7. *Protocol stack of DPWS (Zeeb, et al. 2007).*

The explanation of security for DPWS devices will be extended within the Security section of this chapter.

2.1.4 Web Services in Industrial Systems

During recent years the use of WSs in industrial systems has acquired special relevance due to the fact that manufactures and integrators are switching to Service Oriented Architecture (SOA) paradigm. In SOA the concept of service is used as an abstraction which encapsulates some functionality, in this way, an industrial process can be encapsulated and exposed as service. This service can be published, located and invoked. Services are designed to be modular and self-contained (Delamer and Martinez Lastra, Loosely-coupled Automation Systems using Device-level SOA 2007).

Under the SOA pattern, business and industrial processes can be interconnected, composed and their execution choreographed in a low-cost, flexible, reusable and scalable way (Komoda 2006). To fulfill these principles, the WS framework is used for developing SOA-based systems (Erl 2009).

A number of research projects have demonstrated the feasibility of using WS in the industrial domain. The SIRENA project leverages SOA to interconnect seamlessly em-

bedded devices across the industrial, telecommunication, automotive and home automation domain. The core technology of the SIRENA infrastructure is the DPWS, previously introduced. The industrial demonstrator of the SIRENA project is a fully functional dose maker, which components are exposed as services by using DPWS devices that are used to coordinate and communicate the demonstrator components (Bohn, Bobek and Frank 2006).

Another project that has boosted the use of WS in the industrial domain is *Service Oriented Device and Delivery Architecture* (SODA). Its main objective was to create a service-oriented ecosystem, for high-level communication between embedded devices. During this project tools and frameworks, for enabling WS in resource constrained devices, were created. As its predecessor project SIRENA, SODA was developed around the use of the DPWS devices (de Deugd, et al. 2006).

The SOCRADES project proposes an integration architecture for the manufacturing shop-floor to enable networked intelligent machines to expose their functionality as WS. With this, horizontal and vertical integration across different domains is possible (Moreira Sa de Souza, et al. 2008), as shown in Figure 2.8. The most important enhancements in the manufacturing domain achieved with this project are (Cannata, Gerosa and Taisch 2008):

- ***Reconfigurability***: discovery and composition of services removes the use software modification every time the system changes.
- ***Interoperability***: under the SOA paradigm, heterogeneous devices are loosely coupled and services can be developed on distinct platforms
- ***Scalability***: especially important for manufacturing, pieces of equipment can be removed or attached to the system spontaneously
- ***Reusability***: once the devices functionalities are encapsulated as services, they can be reused by any SOA-enabled system
- ***Maintenance Optimization***: Failure detection among sensors, resources and controllers can be done in the same fashion. Information granularity increases since data arrives from low level resources like sensors and actuators.

Within the SOCRADES project, pilots with embedded industrial controllers having WS capabilities were deployed and it was possible to orchestrate devices processes with their functionality encapsulated as WS (Karnouskos, Bangemann and Diedrich 2009).

The previous projects covered the area of WSs at device level for industrial control applications. In order to address the WS-based monitoring of industrial environments, the Service-bAsed Monitoring for Industrial Ambients (SAMIA) project was developed. SAMIA leverages the event-based capabilities of industrial controllers that implement the DPWS to perform remote monitoring utilizing effectively the network bandwidth. One of the main objectives of SAMIA was to acquire information directly from the sensor, actuator or piece of equipment that produces the data without going through centralized data-gathering systems. Real industrial scenarios, in the manufacturing and utility sector, were used as proof of concept. (SAMIA 2009).

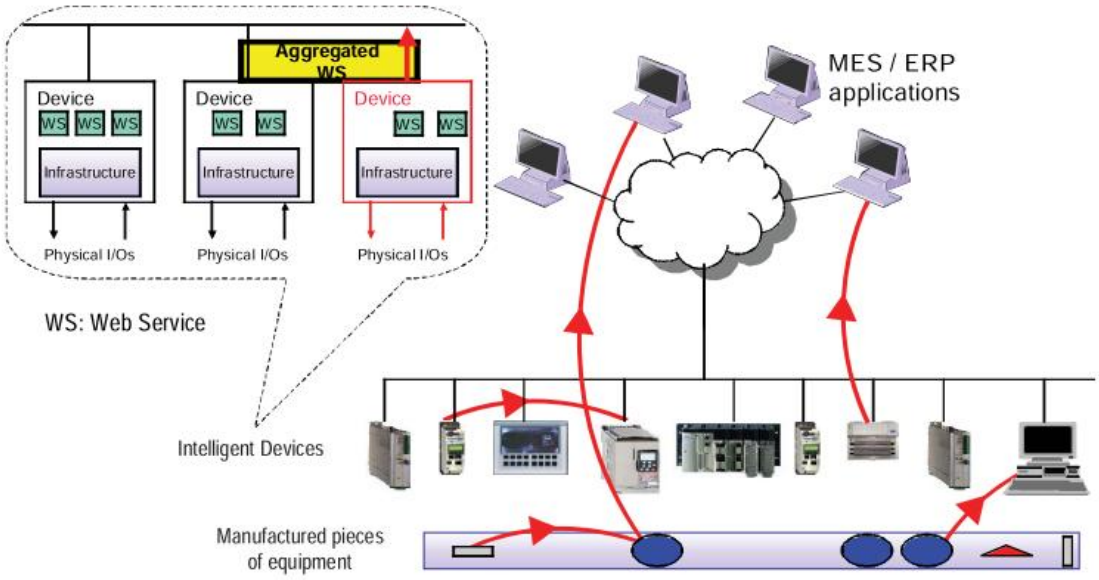


Figure 2.8. *WS at device level in the shop-floor (SOCRADES 2009).*

Other academia efforts that empower the use of WS in the industrial domain, deal with the rapid reconfigurability of manufacturing systems using Semantic WSs as a tool to incorporate knowledge automatically in the devices that compose the system (Lastra and Delamer 2006). Frameworks like TiCS has been developed to facilitate the development, deployment and invocation of time-constrained WS, so that real-time constraints of manufacturing processes can be addressed when using WS in a SOA fashion (Mathes, Heinzl and Freisleben 2008). Methodologies that facilitate the systematical detection of WS operations in manufacturing systems, have been researched, to allow equipment manufacturers, systems integrators and end users to understand and exploit the whole potential of using WSs, in a service oriented fashion, within industrial systems(Villasenor Herrera, Nieto Lee and Martinez Lastra 2010).

One tangible example of the WS technology in commercial applications for the industrial domain is the OLE for Process Control Unified Architecture (OPC UA), which is the new version of the already known OPC architecture used for collection of real-time data. The OPC UA adopts WS to enable interoperable, cross-platforms communications. It also leverages this technology to provide secure transmission of data and flexibility when configuring the system. The OPC UA makes use of several WS specifications in its communication core (Candido, et al. 2010):

- WS-Discovery
- WS-Inspection
- WS-SecureConversation
- WS-Trust
- WS-Security
- WS-Addressing

2.2 Security

In information security, usually referred as cyber security, a risk exists if there is vulnerability and a threat. A component is vulnerable if damage can be caused on it due to design holes, implementation flaws or fundamental weakness. A threat exists when an attacker tries to use the vulnerability to infringe damage. It is important to mention that damage in the system can be caused also by no intentional actions, for example, by design errors that run on top of the vulnerabilities (Dzung, et al. 2005).

Security in distributed systems focuses on protecting the assets that belong to a computational system from security risks. These assets are varied and they range from physical devices like computers and industrial controllers, to information either storage or in transmission by providing countermeasure mechanisms in order to prevent or eliminate the risks, vulnerabilities or threats(ISA 2007).

Securing a system implies a compromise that is usually reflected in its performance, due to the fact that extra computational efforts are required to protect the information; therefore, the strength of the security mechanisms depends on the relevance of the system to protect and the importance of the data that is secured, for example, in an industrial scenario it might be more relevant to protect a message that contains the configuration parameters of a process than a message that contains the reading of a single sensor. The classification of data depending on the need of protection is known as data sensitivity and is important when considering the security mechanisms that protect it.

When dealing with cyber security, it is common to refer to the Open Systems Interconnection (OSI) model layers in order to point, in a communication stack, where threats, attacks and secure mechanisms can be implemented. During this document section, the OSI model is frequently referred.

2.2.1 Security Fundamentals

This section introduces key concepts in cyber security. For instance, the main goal of cyber security is to ensure one or more security services against attacks. Both concepts are explained next.

Security services

In distributed systems, security services can be provided to ensure safeness of the exchanged messages and the system itself. The most common are the following (IEEE PRESS 2007):

- ***Authentication***: used to prove that the identity claimed by an entity is valid (peer entity authentication) or that the origin of the data is valid (data origin authentication)
- ***Authorization***: grants access to operations on different resources, such as read, write or execute. Different entities have different permissions levels, in order to authorize the use of operations on a resource; the authentication of the entity has to be done prior.

- ***Data confidentiality***: protects the data of being disclosure to an unauthorized entity. The confidentiality can be granted on different layers of the OSI model (connection confidentiality) or at specific fields of data in a sent message (selective field confidentiality).
- ***Data integrity***: ensures that during transmission, the data is not altered by unauthorized entities. Selective field connection integrity warranties integrity of specific fields of data within a message.
- ***Non-repudiation***: proves that a transaction did occur between two entities. For instance, with non-repudiation with proof of delivery, the sender receives a notification of the delivered message so that later the receiver cannot deny the reception of the message.

The previously mentioned security services can be provided on different layers of the OSI model as shown on Table 2.1, it is important to highlight that all the security services can be implemented in the layer 7 of the OSI model, the Application layer.

Table 2.1. *Relationship of Security Services and Layers 1-7 (IEEE PRESS 2007).*

Service	1	2	3	4	5	6	7
Peer entity authentication			X	X			X
Data origin authentication			X	X			X
Access control service			X	X			X
Connection confidentiality	X	X	X	X		X	X
Connectionless confidentiality		X	X	X		X	X
Selective field confidentiality						X	X
Traffic flow confidentiality	X		X				X
Connection integrity with recovery				X			X
Connection integrity without recovery			X	X			X
Selective field connection integrity							X
Connectionless integrity			X	X			X
Selective field connectionless integrity							X
Nonrepudiation of origin							X
Nonrepudiation of delivery							X

Types of attacks

The security of a distributed system can be compromised by attacks, which can be done in any of the layers of the OSI model. This implies that an attack in a specific layer, can compromise the security mechanisms used in above layers. The basic security attacks are (IEEE PRESS 2007):

- ***Eavesdropping attacks***: in these attacks, the network communication messages are intercepted and the exchanged information is disclosure. It can be performed in different layers, for example by sniffing in the network layer or by physically wiretapping in the physical layer. One very common technique to execute this attack is the Man in the Middle attack, where the attacker positions between two

communicating end points in order to intercept the exchanged messages, disclosure them and even modify them.

- ***Logon Abuse Attacks***: it bypasses the authentication and authorization processes in order to have not authorized privileges on resources.
- ***Spoofing Attacks***: an entity pretends to have a fake identity in order to have rights over resources. A simple example is the IP spoofing, where an attacker modifies its IP, in the transport layer, and then gain access in a system which trusts in the fake IP of the attacker.
- ***Intrusion Attacks***: by using the network, an attacker targets a specific asset with a known vulnerability. For example with buffer overflow attack a server can respond in an unpredictable way.
- ***Hijacking Attacks***: by using a legitimate entity valid connection, an attacker can gain access to the system. For example, in the session layer, if a connection is left open, by session hijacking an attacker can predict the sequence number used by the Transmission Control Protocol (TCP) to gain control over the session; this attack is known as TCP sequence number attack.
- ***Denial of Service (DoS) Attacks***: the attacker makes exhaustive use of an asset in order to saturate its capacity to serve legitimate entities. The most popular DoS attacks are:
    - SYN Attack: where an attacker sends multiple connection requests without acknowledging the server responses. The server waits for those acknowledges and start saturating its connection requests memory until it collapses.
    - Ping of Death: where a ping request is sent with a larger size than expected making the system to crash.
- ***Application Level Attacks***: They exploit vulnerabilities in the application layer by using intrusion attacks. Common examples are Structured Query Language Injection and Cross-Site Scripting.

2.2.2 Security Mechanisms

Security services are possible through the implementation of security mechanisms. In most of the cases, security mechanisms rely heavily in cryptographic methods. The practical implementation of the security mechanisms is done by applying security protocols, which are implemented at specific layers of the OSI model. The security mechanisms can be grouped into different categories which are detailed next:

Encryption mechanisms

Encryption mechanisms are used to storage and transmit data securely. By applying encryption algorithms, the input data, usually called plaintext, is transformed into ciphertext, which is unintelligible information for all the observers except for the intended receiver; this is possible by using encryption and encryption keys. The security strength

of the ciphertext depends in the used algorithm and the size of the encryption key. Modern encryption algorithms are classified into symmetric and asymmetric algorithms (Mogollon 2007), which are briefly introduced next.

In symmetric encryption algorithms, also called, secret key encryption, the encryption key is the same that the decryption key. This means that both, the sender and the receiver, must exchange securely the key and it must be kept secret. If the key is disclosure to an unauthorized entity, the communication channel is not secure anymore. If individual secure channels are desired, a secret key is required per each channel, a distributed system with n entities, require $\frac{n(n-1)}{2}$ keys, creating complications for the distribution and management of keys when the system scales, due to the exponential growth of the function. Some typical symmetric algorithms are RC4, DES and Advanced Encryption Standard (AES) (Dzung, et al. 2005).

In asymmetric encryption, also known as public-key cryptography, a pair of keys is created. The public and the private key. The private key must be kept in secret but the public key can be distributed to other entities. It has two essential properties:

- All messages encrypted with the public key, can be just decrypted by the participant that poses the private key. All other participants cannot decrypt the message. As shown in Figure 2.9 this provides unilateral confidentiality to the communication.
- Messages encrypted with the private key can be decrypted only by its related public key. This property enables the verification of the integrity and origin of the message.

Asymmetric encryption algorithms do not require a secure exchange of keys unlike symmetric algorithms but they are computationally more demanding, reducing the performance of the communications. The most used of these algorithms are the RSA, ElGamal and Diffie-Hellman (Mogollon 2007).

There exists the possibility to use hybrid encryption to leverage the advantages of symmetric and asymmetric encryption. By using asymmetric encryption, the two entities agree about a secret key that will be used later to encrypt the messages with symmetric algorithms. By doing this is not necessary to have a secure channel for the secret key distribution plus by using symmetric algorithms the computational demand for encryption/decryption process is reduced compared with asymmetric algorithms (Dzung, et al. 2005).

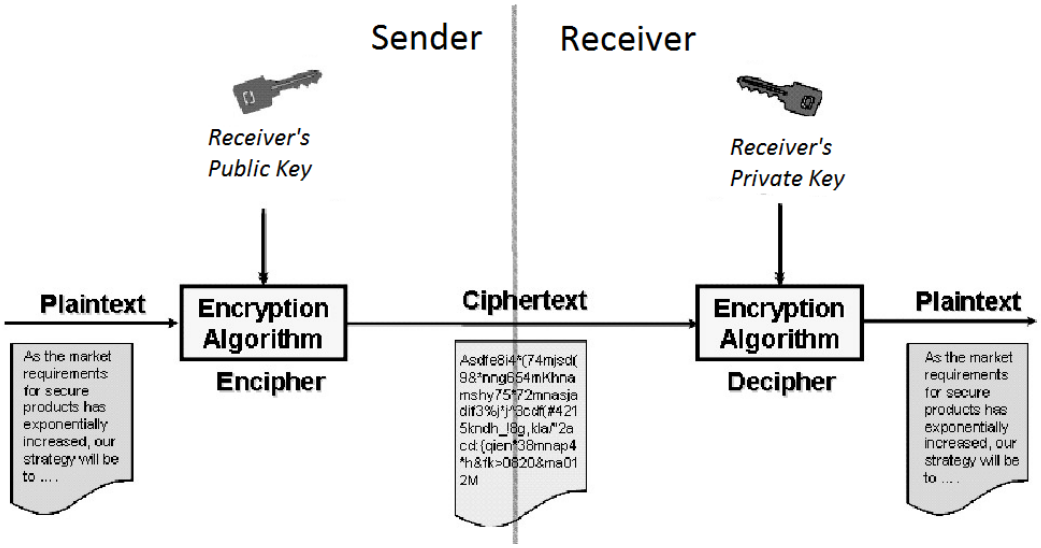


Figure 2.9. *Confidentiality in asymmetric encryption. Modified from (Mogollon 2007).*

Data Integrity Mechanisms

By doing a cryptographic checksum of the message, called hash, it is possible to know if the original data has been altered during a transmission because any change in the data will produce different hash value from the original. Thus if the hash is transferred securely, it is possible for the receiver to compute the message hash and check it with the received hash value in order to confirm the integrity of the message (IEEE PRESS 2007). Hash values are used in digital signatures which are introduced next.

Digital Signatures

Digital signatures are used to proof message integrity and origin. They rely in the use of asymmetric encryption algorithms. The sender uses its private key to encrypt the message's hash, which is then a digital signature. Then the receiver can compute the message hash and compare it with the encrypted hash by decrypting the digital signature with its public key. If they match the message integrity and origin are verified (Dzung, et al. 2005). Digital signatures can be applied to specific elements of an XML document by using XML Signature. The typical algorithm for producing digital signatures is RSA.

Other advantage of using digital signatures is that they can enable authentication services, by using a proper infrastructure and methods to correlate signatures and identities. Non-repudiation service is another possibility because once a document is signed with the private key, the participant that signed it cannot deny the emission.

Authentication Mechanisms

The authentication service can be provided by mechanisms like passwords, messages encrypted with secret keys shared by authentic participants or by using trust infrastructures like:

- Public Key Infrastructure (PKI) which allows service consumers to validate the identity of a service provider by using a trusted party (Certificate Authority) which corroborates the validity of digital certificates. A digital certificate is composed by data and a digital signature. In the data part, a public key and an associated identity are included. This is the key point that allows a digital certificate to establish authentication since it relates a key with an identity (IEEE PRESS 2007).
- Federated security infrastructure which allows the separation between the authentication and authorization processes from the services that clients consume by providing a central repository of IDs. This facilitates the management of the credentials and enables secure interoperability between different organizations. SAML assertions is a technology used to implement federated security (Hatala, Mey and Shah 2005) and is described later in this section.

Access Control Mechanisms

These mechanisms are used to control the access to services and resources offered in the network. Once an entity has been authenticated, its rights over the services and resources are determined by using access policies or checking the authorized capabilities in authorization lists. Another example of access control mechanisms are firewalls (IEEE PRESS 2007), which are going to be detailed on the next subsection.

Intrusion Detection Systems (IDS)

An IDS discovers attacks by analyzing abnormal behavior of the network traffic and by identifying known attacks profiles. Common indexes to be analyzed are type, content, frequency and path of the exchanged messages (Dzung, et al. 2005).

2.2.3 Tunneling Protocols and Firewalls

Tunneling protocols and firewalls are security mechanisms widely used in practice. The vast amount of research and commercial implementation makes them good candidates to be evaluated for protecting industrial environments that are monitored with WSs.

Tunneling Protocols

A tunneling protocol encapsulates the payload data in order to transport it in a secure manner. They can operate at different layers of the OSI model. The most popular are introduced next (IEEE PRESS 2007):

- ***Point-to-Point Tunneling Protocol***: is a layer 2 tunneling protocol that allows authentication and the encryption of data for remote-access.
- ***IP Security (IPSec):*** works on layer 3. It is a set of open standards that offers a broad variety of encryption algorithms in order to provide confidentiality, integrity and authentication. It is widely used to deploy Virtual Private Networks (VPNs) which are introduced later in this subsection.

- ***Transport Layer Security/ Secure Sockets Layer (TLS/SSL)***: is a Transport[1] layer tunneling protocol, implemented in most web browsers to protect HTTP messages. These protocols allow the client and the server to agree about a suit of cryptographic methods to establish a secure channel in the handshake stage. In the Record Layer stage, the protocols derive a symmetric key to be used in symmetric encryption algorithms for data protection (Gupta, et al. 2005). SSL/TLS also provides server authentication by using digital certificates.

Tunneling protocols (especially IPSec) can be used for building a VPN. A VPN allows users or sites to connect through the Internet to a remote private network. All the information is tunneled thus provides data confidentiality, integrity, authentication and access control. The working principle of a VPN is simple, the payload is encapsulated with a secure tunneling protocol when it enters the tunnel endpoint and deencapsulated when it goes out from the tunnel. Depending on the authorization policies, once an entity is virtually connected to a remote network through a VPN, it can make use of the networked resourced (IEEE PRESS 2007). VPNs can be done site-to-site or client-to-site, Figure 2.10 shows illustrates the former one.

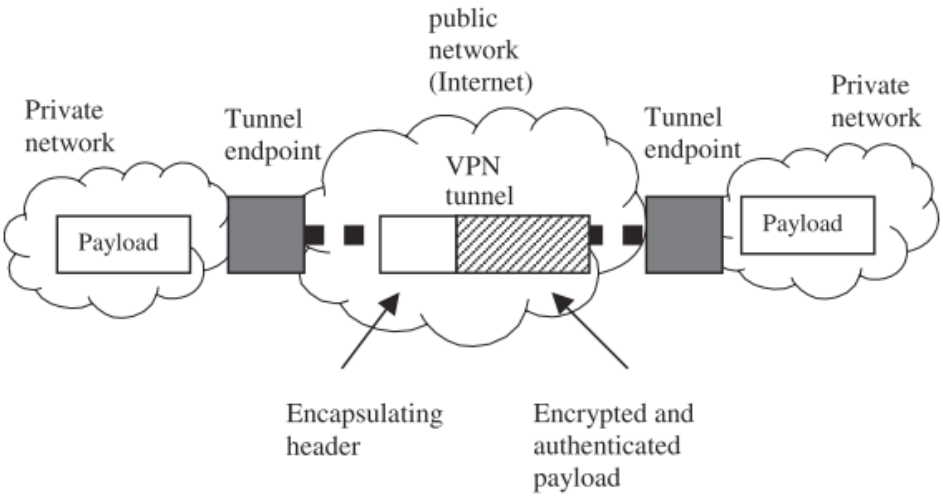


Figure 2.10. *VPN elements and operation. Modified from (IEEE PRESS 2007).*

Firewall

A firewall is an element that is placed between two networks to allow or block the traffic between them, according to some security policy. An important assumption when designing security with firewalls is that all entities behind the firewall are trusted. In case this trust condition cannot be granted, internal firewalls are deployed to isolate the trusted network from the non trusted entities.

It is common to use two firewalls (embedded in the network routers) to create a restricted access network called demilitarized zone (DMZ). The DMZ zone can be accessed, under security policies, from non trusted external networks. With this architec-

[1] In practice TLS/SSL is applied on the Tranport layer or any of its upper layers, thus it can be considered also an Application layer mechanism.

ture it is possible to protect internal entities from outside attacks and allow entities in the DMZ zone to provide services to external clients (IEEE PRESS 2007). The layout is illustrated in Figure 2.11.

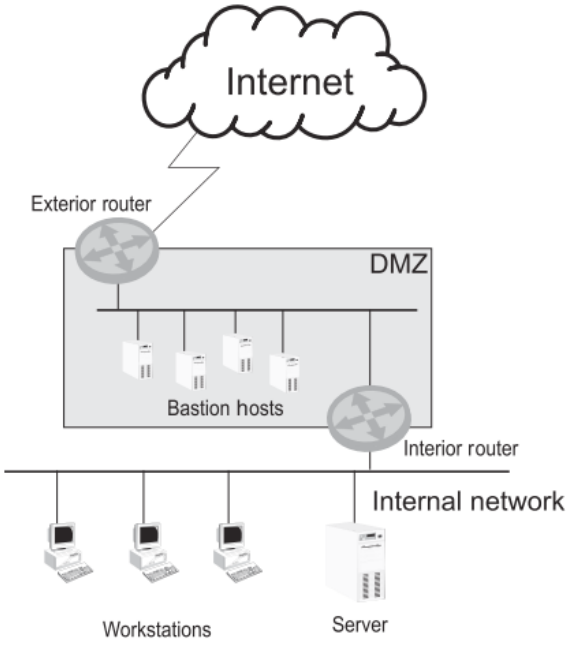


Figure 2.11. *Typical firewall arrangement (IEEE PRESS 2007).*

Most commonly firewalls filter the traffic by applying policies that deal with the layer 3 (Network) and layer 4 (Transport) of the OSI model, nevertheless there exits Layer 7 (application-level) firewalls, commonly called, application-level gateways, which allow higher level policies for filtering. This gives the designer great flexibility for establishing security policies at the application layer. Layer 2 (Data link) firewall are used to control traffic depending on security policies on the data link layer. Their main advantage is that they can be installed on demand without segmenting the network, thus network devices don not need to be reconfigured since they preserve their IP addresses. Layer 2 firewalls are very useful to prevent IP spoofing attacks by checking that the IP – MAC addresses are consistent (IEEE PRESS 2007).

Since all the traffic need to pass through the firewalls, they are usually a bottleneck point and the communications performance is decreased when the traffic overcomes the firewall capacity. Another drawback is that firewalls have difficulties when working with tunneling protocols in VPNs like IPSEC since some data fields are encrypted, they cannot be evaluated in the firewall security policy(IEEE PRESS 2007).

2.2.4 Web Services Security

WS security relies pretty much in technologies that secure XML messages since they are the core format of SOAP messages. WS standards, which employ these technologies, have been developed by different organizations to cover gaps between current security standards and WS (IEEE PRESS 2007). These standards are grouped within the

Security category of the WS standards stack. The technologies and WS standards that allow the implementation of security in WS are shown in Figure 2.12.

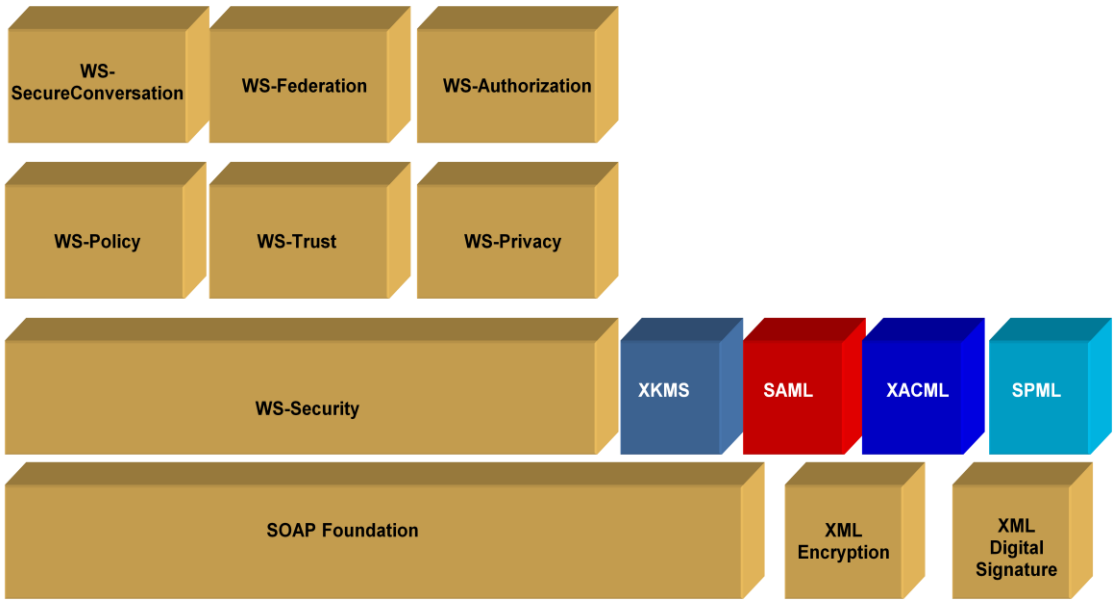


Figure 2.12. *WS Security Standards (WS-I 2004).*

XML Encryption

XML Encryption provides the rules and syntax for encrypting arbitrary data in an XML document. With XML encryption different elements of an XML document can be encrypted independently of each other. It is highly interoperable with XML-related applications, thus it can be used to provide confidentiality in SOAP messages and therefore WS.

The XML encryption syntax includes the encrypted data structure which holds the encryption elements like: EncryptionMethod, KeyInfo or CipherData. Some of these elements can be omitted which gives flexibility to the applications that use this technology.

XML Signature

XML Signature provides a set of rules and the XML syntax to encrypt, process and verify digital signatures of arbitrary data fields of an XML document. An XML digital signature is the encrypted hash value of an XML element or document. It provides integrity and proof of origin to WS messages.

XML documents can have the same information with different structures thus for the same information contained in XML documents, different digital signatures can exist. This complicates the integrity verification process, so special considerations have to be taken into account. One way to solve the previous is by converting the received document to its canonical form by applying XML canonization.

A key feature of XML signature is that different elements of an XML document can be signed by different parties, allowing a message to travel among different participants independently of their trust domain. Thanks to its interoperability with XML technologies, it is widely used by WS-Security standard, described lately.

The XML signature syntax includes elements like: *SignedInfo, SignatureValue, KeyInfo* and *Object* for any data that might be required for the signature process. Flexibility is possible since some of these elements can be omitted.

XML Key Management Specification (XKMS)

XKMS is a specification that allows the management (distribution, register, revocation) of public keys in a WS fashion. It is analog to PKI but it simplifies it since it defines programming interfaces that can be used by WS as shown in Figure 2.13. The WS functionalities exposed by XKMS are: Registration of public key, key revocation, key recovery, location of the public key and key validation. XML Signature and XML Encryption are the only cryptographic technologies required for implementing XKMS.

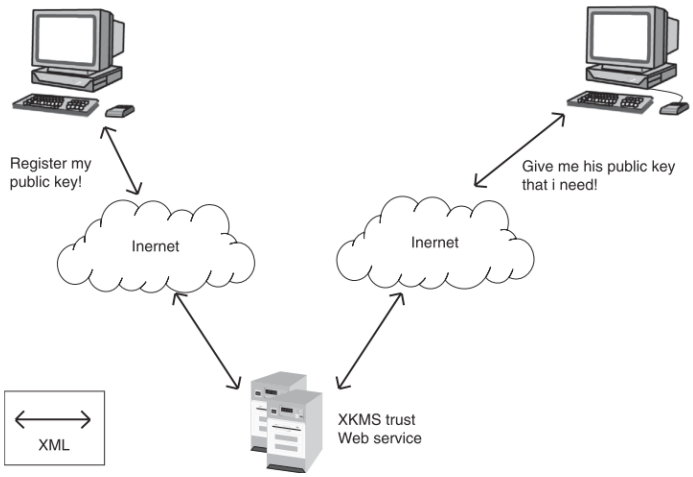


Figure 2.13. *XKMS topology (IEEE PRESS 2007).*

Security Assertion Markup Language (SAML)

SAML is a standard that provides an XML-based framework that allows to exchange security information related to an entity in the form of assertions, so that it can be recognized by all the participants. The assertions are usually referred as tokens. SAML classifies three types of assertions that can be created:

- ***Authentication***: proofs that a participant identity has been authenticated.
- ***Attribute***: states the attributes that are associated with a particular identity.
- ***Authorization Decision***: states privileges over a resource for a participant.

The classical topology includes a service consumer, a trusted security authority, usually called Security Token Service (STS) and a service provider. The consumer requests the STS to return a SAML token that proofs its authentication. Then the consumer requests a service from the service provider by attaching the SAML token received

from the security authority. Finally the service provider validates the SAML token which is signed by the STS and thus protected against tampering.

The main purpose of SAML is to standardize the representation of security data so that it can be processed by different security related services. The mechanisms that are used to exchange the data are out of the scope of the SAML standard. SAML is an enabler to implement federated security architectures which improves the performance of security in distributed systems and reduces the complexity of managing security credentials.

eXtensible Access Control Markup Language (XACML)

XACML is an XML-based standard that outlines the syntax to express general purpose access policies. It also defines a processing model that allows combining individual policies into a single set. The implementation of XACML needs different components like: Policy Administration Point (PAP) where policies are managed, Policy Decision Point (PDP) where access to resources is determined or Policy Information Point where external information can be linked to be used by the PDP. XACML facilitates to implement access mechanisms to WS and thus provides an authorization service.

Web Services Security Standards

They are a set of SOAP extensions which are used to secure WS under different contexts and security models. The WS security standards allow the designers to implement security services without affecting the functional characteristics of the system. Their main purpose is to allow secure interoperability of systems preserving essential characteristics of WS like platform independence, composeability, flexibility and modularity. A key characteristic of WS Security standards is that they provide end-to-end security in the communication, meaning that a message can pass through different participants preserving specific security requirements for each of them.

The WS Security suite is formed by individual and interrelated specifications which are associated in a layered architecture; see Figure 2.12. Thus allowing the designer to mix and implement just the needed components. The WS Security suite main standards are:

- ***WS-Security***: is the basis for other security specifications since it provides functions that enable integrity and confidentiality to WS messages. It relies heavily in the use of XML Signature and XML encryption plus it makes use of other technologies like SAML assertions. Some branches of this standard are (InnoQ 2007):
    - SOAP message Security: provides flexible support for multiple security token formats, trust domains, signature formats and encryption methods.
    - Username Token Profile: Indicates how a WS client can provide Username Token to identify and authenticate the WS provider.
    - Kerberos Binding: specifies how Kerberos tickets should be encoded and attached in SOAP messages when using the WS-Security: SOAP mes-

sage Security specification. Kerberos tokens are common to be used within local networks.
    - SAML Token Profile**:** describes how SAML assertions should be used with the WS-Security: SOAP message Security specification.
    - X.509 Certificate Token Profile**:** defines how to use the X.509 technology with the WS-Security: SOAP message Security specification. It is common in scenarios where business partners need to exchange information.
- ***WS-Policy*****:** it defines a model and syntax to express and communicate general purpose policies related to a WS. Some subspecifications of this standard are:
    - WS-PolicyAttachments**:** describes how a policy can be attached to a WS.
    - WS-PolicyAssertions**:** enumarates a set of general-purpose assertions that might be useful when using a WS.
    - WS-SecurityPolicy**:** used to provide information concerning the WS-Security tokens.
- ***WS-Trust***: describes the model that allows the establishment of trust relations by using security tokens. It provides extensions on top of WS-Security that allows the entities to exchange, issue, renew and validate security tokens. Most commonly the security tokens are expressed as SAML tokens. WS-Trust is a key enabler of a federated security system.
- ***WS-Privacy***: it specifies the model to integrate statements about privacy in WS. It makes use of WS-Policy to communicate the statements. It describes how WS-Trust can be used to evaluate the privacy claims and it uses WS-Security to support cryptographic methods and tokens.
- ***WS-SecureConversation***: it is used to establish a secure session and this helps to reduce the overhead of exchanging multiple independent messages because a unique Security Context Token (SCT) is used to authenticate all the messages exchanged within the session. This protocol improves the performance of systems that require constant exchange of messages.
- ***WS-Federation*****:** it describes a federated model for WS across different trust domains by merging the use of WS-Security,WS-SecureConversation and WS-Trust. The standard specifies protocols for managing transactions and defines security precautions in a federated system. It is designed to be highly composable with WS standards within the category of Reliable Messaging.
- ***WS-Authorization***: it indicates how claims contained in tokens should be structured, interpreted and processed in order to permit or deny access to WS. It is used as an access control mechanism.

In the common model, when implementing the WS Security standards suite, a WS provider requires a set of claims to be fulfilled before providing the service. This set of claims is expressed as a policy by using WS-Policy. Common claims would be identity credentials or access permissions. Then the service consumer requests those claims to a

security authority, the STS, which is exposed as a WS. The STS returns the requested claims in the form of a security token. This security token is signed by the STS and can be verified by the WS provider. This messaging interaction is illustrated in Figure 2.14.

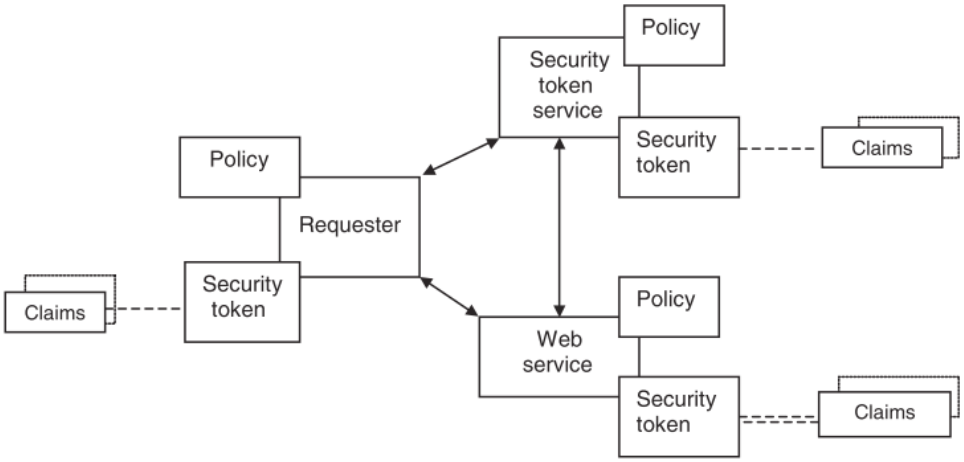


Figure 2.14. *General WS Security messaging (IEEE PRESS 2007).*

2.2.5 DPWS Security

The DPWS specification has a section dedicated to describe how security should be implemented, the main guidelines and points are summarized in this subsection. It is also given a review of the research efforts that attempt to implement security mechanisms in the DPWS stack.

DPWS 1.1 Standard

In the security section of the DPWS 1.1 standard, it is recommended a minimal set of technologies and mechanisms to enable secure communications between devices and clients. The standard is flexible and allows the devices to support alternative security mechanisms and profiles in order to address the application requirements. The use of other security profiles may by indicated by means of WSDL or policies (using WS-Policy). It is responsibility of the implementer to take care of composeability of standards for each security profile in order to achieve interoperability (OASIS 2009) .

The DPWS security focuses on services like: authentication of devices, integrity and confidentiality of messages exchanged between the devices. To achieve this, the standard proposes model composed by two parts:

- ***Message-level signatures for UDP WS-Discovery traffic***: with WS-Discovery Compact Signatures a client can proof integrity of the received discovery messages and proof the identity signed by the sender. This mechanism requires the use of WS-Security to generate the cryptographic signature.
- ***Transport-level encryption for metadata and control traffic***: in order to establish a secure point-to-point communication channel TLS/SSL is used. This allows authenticating the identity of the participants and ensures the integrity and confidentiality of the exchanged messages. Once the authentication has been proved, the HTTPS protocol is used to exchange the messages.

The standard recommends that devices use x.509.v3 certificates as their credential. This credential can be used to sign WS-Discovery messages and for establishing the TLS/SSL channel. Devices can authenticate a client by using their x.509.v3 certificates obtained in the TLS/SSL negotiation or can require their username/password credentials once the secure channel has been established. The general diagram of this model is illustrated in Figure 2.15.

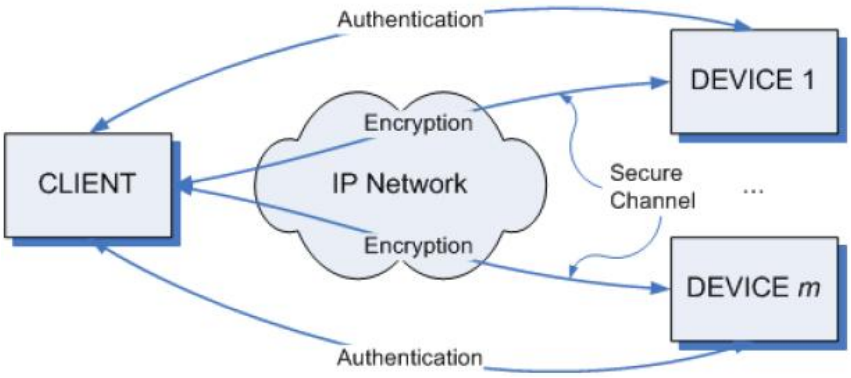


Figure 2.15. *DPWS Standard security model (OASIS 2009).*

Research efforts

The SODA project, already introduced, proposed a security framework to enable secure messaging and access control among networked devices. The framework provides an extension for the DPWS standard. Some of the main features are (Martinez, et al. 2008):

- Leverage the message-level security properties by using WS-Security
- The security level implemented by a device depends on its computational capabilities
- The authorization and authentication methods can be performed locally on the device or in trusted elements
- Security management is done remotely

The architectural elements of the SODA security framework, see Figure 2.16, are:

- Devices: They use the DPWS stack and can request or provide a service.
- Authentication Server: Trusted entity that authenticates devices by sending them a token.
- Authorization Server: Trusted entity check the client rights and grants or denies access to a service
- Security Management Application: Tool used to manage security configurations on servers and devices.

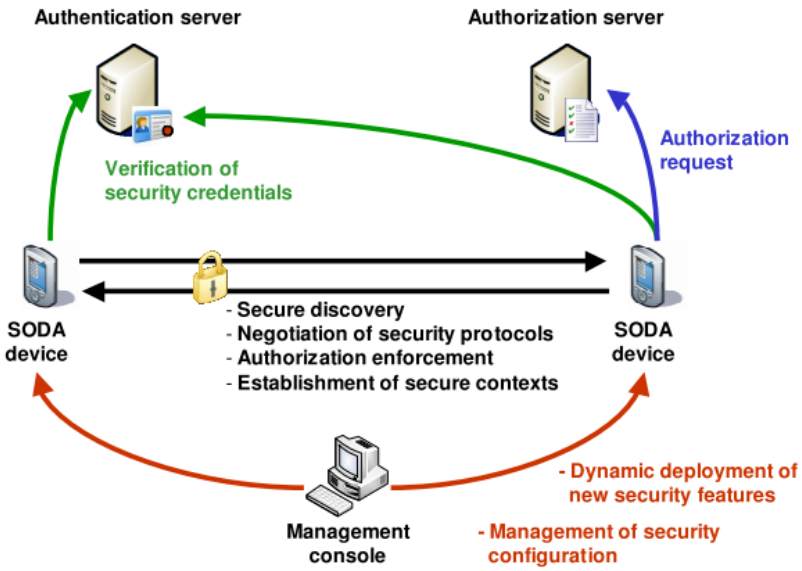


Figure 2.16. *Architecture on the SODA security framework (SODA 2007).*

In order to accomplish with the flexibility stated by the SODA security framework and preserving interoperability with the DPWS stack, SODA proposes that the embedded devices arrange their components as depicted in Figure 2.17. The Security Subsystem interfaces with the trusted entities that provides the authentication and authorization services, in case they do not exist, local authentication and authorization is used. The security subsystem also interfaces with the configuration tool and is responsible of handling authorization and authentication cache memory. The security handler processes securely those messages exchanged between devices by applying the already chosen security mechanisms (Martinez, et al. 2008).

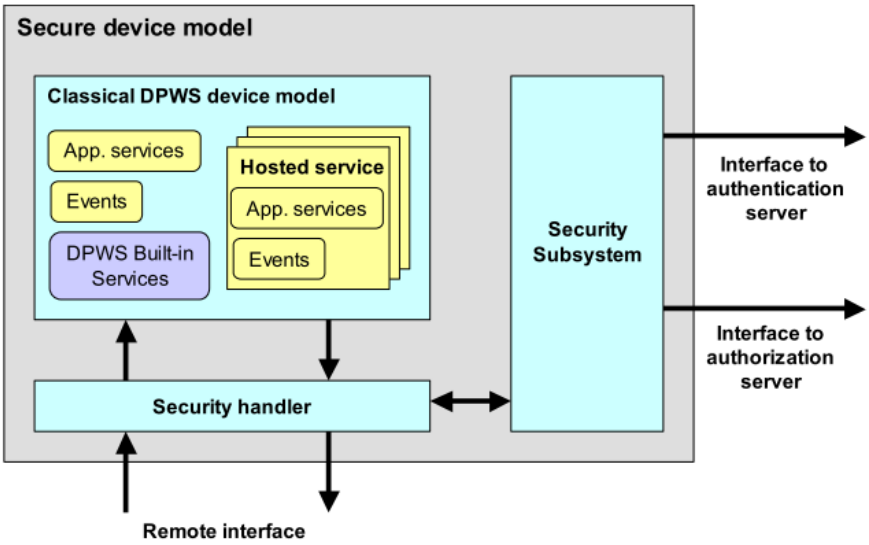


Figure 2.17. *DPWS-based device with security services support (Martinez, et al. 2008).*

An alternative approach for a security framework for DPWS compliant devices, proposes the use of symmetric algorithms for messages encryption, trying to avoid asymmetric algorithms due to the computational effort they request from embedded devices. This approach suggests that the security configuration of the devices should be

done before they are introduced them in deployed network. The encryption key should be renewed periodically by using a second encryption key used for configuration purposes. In this framework the signing of messages is done with detached signatures with the aim of improving performance but sacrificing compliance with WS-Security (Hernandez, et al. 2009).

Another research project, which started recently, is WS4Dsec – Reliably Secure Web Services for Devices founded by the German Research Foundation. The main derivable of the project will be a toolkit that allows the automatic generation of code to reduce the complexity of implementing reliable and secure communications between DPWS devices (WS4DSEC 2010).

Tools that facilitate the integration of security properties in deployments that use DPWS devices have been developed recently. For example, APEL is an application that was developed to design the orchestration of services meanwhile allowing the user to specify security properties for each of the communication links. All of this within the context of an industrial plant where services are hosted by DPWS devices. With this tool it is possible to generate the WS that include the code which supports the security properties decided at the design phase (Chollet, Lalanda and Bottaro 2008), see Figure 2.18.

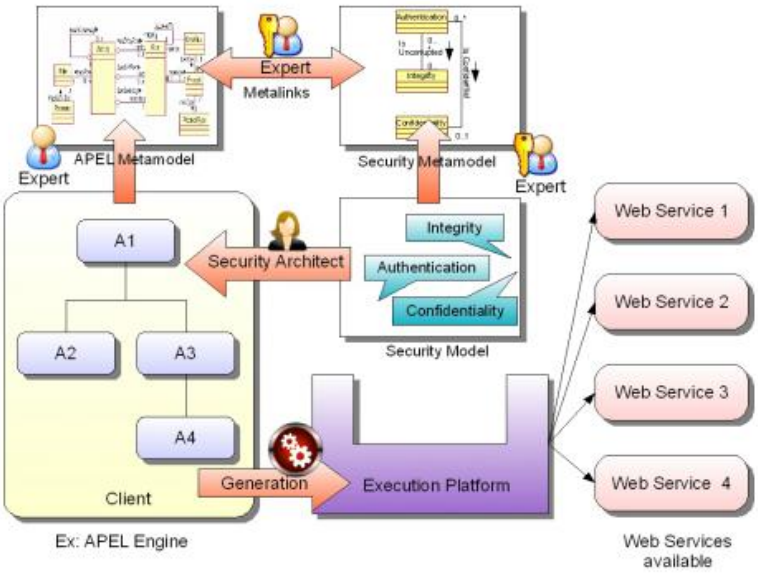


Figure 2.18. *Secure orchestration of WS with APEL (Chollet, Lalanda and Bottaro 2008).*

In order to provide an authorization mechanism for services hosted in DPWS devices, (Suomalainen 2007) proposes an architecture with a middleware-based authorization. The key element is an access control component which makes the authorization decisions. With this architecture is possible to have fine-grained and flexible end-to-end authorizations. Permission attenuation is used when a service is composed by other services. This mechanisms works by transferring and concatenating the security credentials of the entities involved in the composed service.

Another approach that deals with authorization in DPWS devices, is the one proposed by (Muller, et al. 2010) where XACML policies are used to grant access to services. They propose the auto generation of policies templates based on the descriptions provided by the DPWS devices. Within this document they also suggest the use of DPWS proxies to enable the remote discovery of services and the use of a DPWS firewall to limit the visibility and access to the hosted services. This is done by implementing the PEP functionality of the XACML framework.

2.2.6 Information Security in Industrial Systems

Industrial systems are evolving constantly because of the adoption of new communication technologies. These changes raise new vulnerabilities and requirements that the field of industrial information security has to cope with. Industrial deployments used to be protected under the schema of *Security by Obscurity* where the industrial facility used to be isolated from untrusted networks and technical knowledge about the protocols and equipment was unknown by the attackers. These assumptions are not anymore true for industrial systems because nowadays they use open communication standards and use public infrastructures for data transmission like Internet (Dzung, et al. 2005). *Defense-in-depth* is used nowadays to provide security in industrial facilities by using different layers and mechanisms. The assets exposed in outer zones are less secure that those in inner zones. In this way, critical infrastructure like industrial components should be located within the inner zones (Polk, Malkewicz and Novak 2010). Figure 2.19 exemplifies how different, of the previously described, security mechanisms can be used for securing an industrial system.

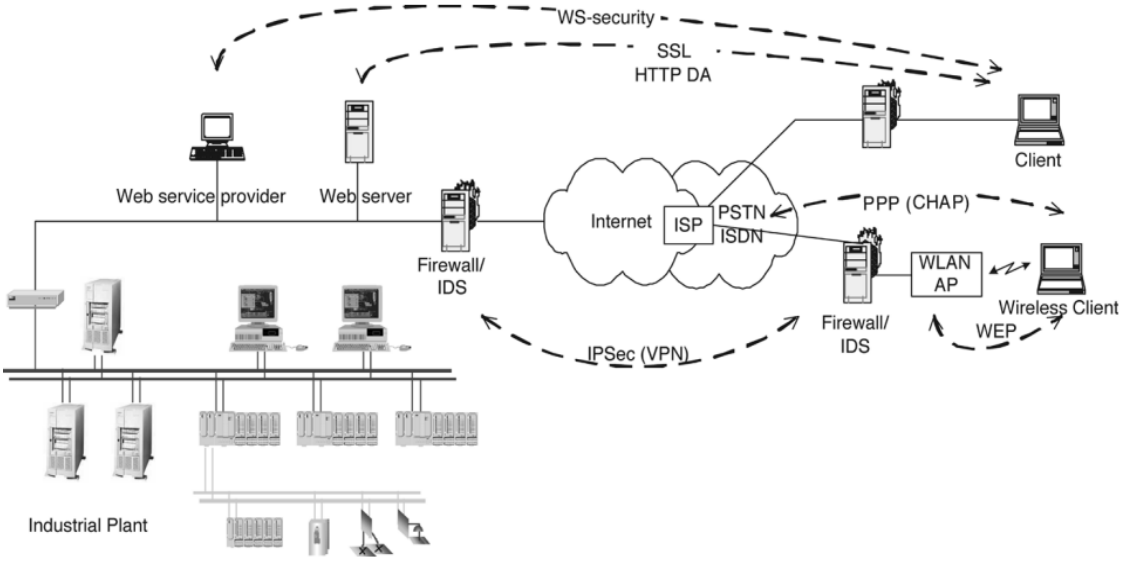


Figure 2.19. *Typical security mechanisms in an industrial system (Dzung, et al. 2005).*

Security in Automation Protocols

Standard automation protocols usually address authentication and authorization services. In (Dzung, et al. 2005) a survey of the security aspects for these protocols is presented. It is summarized and extended with other protocols next:

- ***OPC***: security is optional and addresses access control. It is based on the security model of Microsoft Windows. Credentials in form of tokens are checked in an access control list to grant access to secured objects. Confidentiality and integrity in data transmission is not covered.
- ***OPC UA***: it describes a model to apply different levels of security depending on the application. The security can be applied in three layers: The transport layer, to exchange messages in a secure way by encryption and signatures. The communication layer deals with confidentiality, integrity. The application layer concerns the authentication and authorization services. The system implementers can use this model to select a mechanism from a pool of WSs security specifications. TLS/SSL is the default technology (Armstrong and Hunkar 2010).
- ***MMS***: this is an application layer protocol used to exchange data between PLCs. It allows access control based on simple username/password authentication. Secured objects have access control lists that grants the requester with operation privileges like create, read, write, execute, delete or modify. When used on top of TCP/IP, security mechanisms like IPSec or TLS/SSL can be used.
- ***IEC 61850***: stipulates that participants should authenticate in order to obtain operation privileges. Simple user id and password are sent in each request. It is based on MMS, so as already mentioned, it can use TCP/IP security mechanisms.
- ***ICCP***: this protocol is used for the communication of electric power facilities, substations and control centers through a Wide Area Network (WAN). Originally is not designed with security mechanisms. It runs above MMS using TCP/IP so it can implement the security mechanisms of these technologies.
- ***Field buses***: field buses are designed to be fast and deterministic. It is considered that attackers cannot tamper or eavesdrop field buses messages since that requires physical access to the wiring. Because of these two reasons traditional field buses protocol does not provide security services. Nevertheless field buses gateways that transport data among untrusted network can include security services.
- ***Industrial Ethernet:*** field buses that use the TCP/IP protocols leverage the security mechanisms already existing for this stack. For example in (Hosoya and Miyata 2010) a security architecture is proposed to manage and implement security credentials and IPSec in FF-HSE a protocol of the Fieldbus Foundation.
- ***Wireless for Automation***: geographical distributed systems used in the utility sector, need to use wireless technologies for data and command transmission.

Modern wireless data services include security services. Radio jamming can affect considerably wireless communications.

Security Standards for Industrial Communication Systems

Nowadays different initiatives have been working on creating guidelines, standards and regulations with the aim of improving the information security of industrial systems. The main differences between these initiatives are the participant organizations, their goals, the geographical area they cover and the industry scope (Naedele 2005). Those with bigger momentum are introduced next:

- ***ISA 99***: the ISA99, *Industrial Automation and Control System Security* standard is currently composed by two parts. The Part 1 (approved since 2007) provides the foundation by introducing security concepts, terminology and models. The part 1 provides an assessment of cyber security tools, technologies and countermeasures that can be used in the scope of industrial automation and control systems. The part 2 (approved on 2009) focuses on integrating IT security on an existing industrial infrastructure by using a Cyber Security Management System (CSMS) framework. This part proposes cyber security risk assessment methods, security policies, selection of security countermeasures and guide the implementation and monitoring of the CSMS (ISA 2010) as depicted in Figure 2.20.
- ***IEC 61784*:** with the IEC 61748, security issues are addressed for field buses protocols, especially those that work on Ethernet networks by defining security profiles (Naedele 2005).
- ***NERC1300*:** the purpose of this standard is to reduce risk on bulk electric systems by protecting the cyber assets of the system. It provides rules to identify critical cyber assets, facilitate security management and control personal training. Registered power operators should comply with these rules by 2010(Mahboob and Zubairi 2010).

Figure 2.20. *ISA 99.00.02 Cyber Security Management System. Modified from (ISA 2008).*

Security in SCADA Systems

SCADA systems are an essential component for controlling and monitoring the industrial infrastructure of an organization or country. They are widely used in the process sector like pipelines control, refineries, chemical plants, in the utility sector like water plants and in some manufacturing areas (Krutz 2006). Originally they were designed to work on private networks and secure contexts and thus designers left aside security mechanisms. When SCADA systems started to use Internet for communicating to remote locations security vulnerabilities raised. The most common security issues of SCADA systems are (Mahboob and Zubairi 2010):

- ***Poor security policy in the organization***: security policies are not forced to be implemented as they should.
- ***Lack of layered defense***: usually just external attacks are considered without considering insiders.
- ***Lack of access log***: backup of the access logs is not done, neither analyzed.
- ***Internet based SCADA***: Accessing SCADA system from Internet opens possibilities for attacks.
- ***Non-related software on PCs***: computers in the SCADA system might contain insecure or non stable software that can result in unpredictable function of the equipment.
- ***Control software not exhaustively reviewed***: due to the fact that beta versions are usually not tested by a big community.

Improving the cyber security of a SCADA system can be done by keeping SCADA computers isolated from external networks like Internet. If they require to be exposed then security policies should be enforced on them. Wireless communications should be encrypted. The execution of remote commands should be limited. An important layout to protect SCADA systems is to use defense-in-depth. In Figure 2.21 this layout is shown where the protection mechanisms encompass firewalls, IDSs and antivirus software for the PCs. It is important to use zones privileges to grant different operation permissions for the users depending on their zone. A more sophisticated mechanism will block a combination of remote commands that can compromise the stability of the system. Security can be enforced by the use of authentication credentials like passwords (Mahboob and Zubairi 2010).

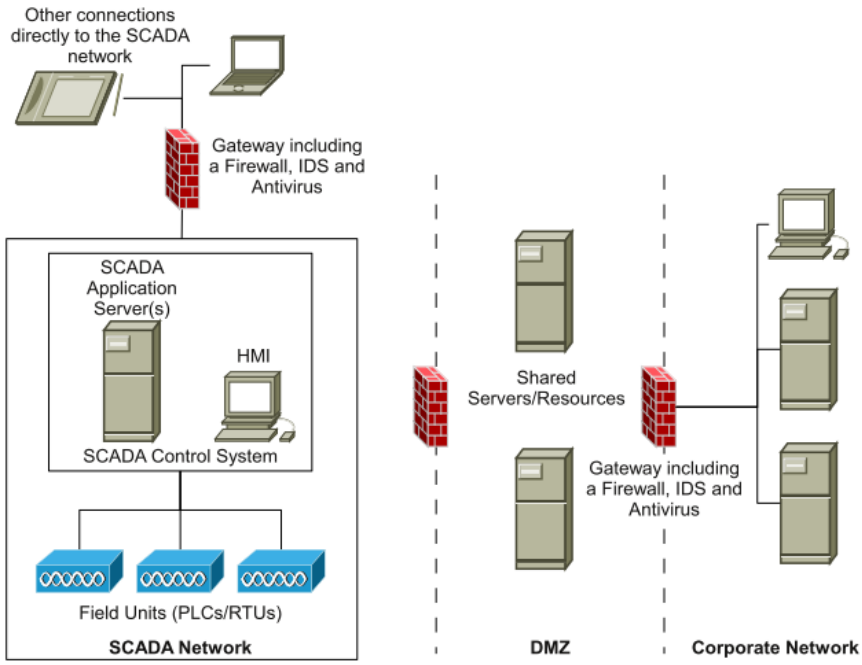


Figure 2.21. *Defense-in-depth for a SCADA system (Slay and Miller 2007).*

Security in industrial networked components

Tangible implementation of security in industrial components is thanks to the OPC UA initiative. OPC UA servers describe its capability and description of security requirements through a Server Profile (Oda 2006). As it was already mentioned (in the Security in Automation Protocols section) OPC UA implements security in three layers by using different specifications belonging to the WS security.

A feasibility evaluation in embedded industrial controllers when implementing different encryption algorithms was done in (Delamer and Martinez Lastra, Information Security for Reconfigurable Manufacturing Systems using Networked Embedded Controllers 2006). The tested controller has a processor with 2MB of RAM and runs at 100MHz. By analyzing the results, shown in Table 2.2, the authors conclude that small configuration files and programs transfer can be secured in their transmission if the application can tolerate delays in the order of seconds.

Table 2.2. *Latency measurements for encryption algorithms in an embedded industrial controller (Delamer and Martinez Lastra, Information Security for Reconfigurable Manufacturing Systems using Networked Embedded Controllers 2006).*

Cipher	Data (KB)	Avg Delay (ms)	Cipher	Data (KB)	Avg Delay (ms)
No cipher	1	6	SHA1 Digest – 160bit	1	56
	128	994		128	7 153
DES – 64bit	1	154	HMacMD5 MAC – 128bit	1	25
	128	14 757		128	2 622
Blowfish – 64bit	1	346	HMacSHA1 MAC – 160bit	1	85
	128	9 931		128	7 683
AES – 128bit	1	132	PSS Signature Generate – 1024bit	1	9 401
	128	18 273		128	15 867
Serpent – 128bit	1	388	PSS Signature Verify – 1024bit	1	1 027
	128	42 248		128	7 895
Rijndael 2.0 – 128bit	1	737	ElGamal Public Key – 1024bit	1024bit	4 331
	128	96 785			
MD5 Digest – 128bit	1	20	RSA Private Key – 1024bit	1024bit	8 760
	128	2 685	RSA Public Key – 1024bit	1024bit	8 837

Industrial gateways with secure services have been widely developed since they can improve the security of the industrial system without major layout modifications. The devices which are connected to other networks through these gateways are protected in a security zone. Examples of them are the SCALANCE series of Siemens which offer industrial secure switches for wired and wireless communications. Secure industrial gateways act as conventional firewalls but keeping industrial requirements like robustness, long lifecycle and reliability as high priorities (Harada 2007).

2.3 3G Mobile Communication

3G mobile communication technology enables voice and data services on mobile appliances. It is standardized by 3GPP and is the successor of GSM and GPRS technologies. 3G communications are in widespread use because they introduce high quality services, data transfer speed is increased and the security framework improved, all of these by preserving the reliability and robustness of its antecessors (Prakash and Behera 2010).

3G networks make possible to reach download speeds up to 7.2Mbps (Jang, et al. 2009) which open new possibilities for a broad range of mobile applications. The radio and handoff mechanisms bring robustness to the connectivity of devices in movement, making suitable to use 3G for appliance located in transport systems. Another key point of 3G networks is their area coverage; base stations can provide service to extensive geographical areas, making this technology suitable to deploy applications in rural and urban locations.

Figure 2.22 shows a simplified version of the 3G network architecture, it is composed by four basic subsystems:

- ***UE***: the *user equipment* that uses the 3G services, it can be any handset or appliance that includes a 3G modem.
- ***RAN***: the *Radio Access Network* allow interface wirelessly with the UE and connects with the core network of the mobile operator. It is composed by nodes (base stations) and radio network controllers.

- ***CS CN***: the *Circuit-Switched Core Network* controls the circuit-switched sessions and allows connection to the public switched telephone network (PSTN) which is known as land lines.(Uskela 2003)
- ***PS CN***: the *Packet-Switched Core Network* is in charge of packet-switched sessions and is the responsible of interfacing the cellular network to external IP networks like the Internet.

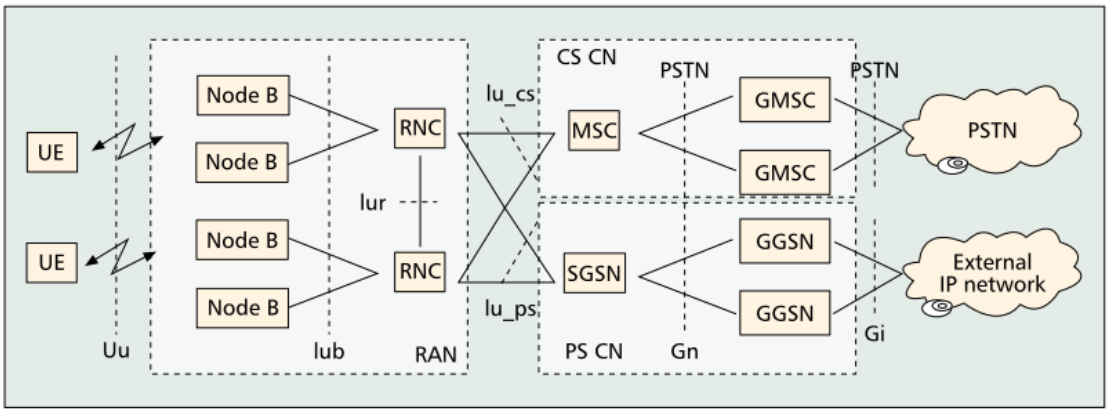


Figure 2.22. *Simplified 3G architecture (Uskela 2003).*

2.3.1 Security

3G security uses some of the security principles used in GSM networks, it improves vulnerable points and provides new advanced security mechanisms. For instance, subscriber authentication, radio interface encryption and the confidentiality of the subscriber identity are preserved and improved. 3G security focuses on protecting the user generated information, resources and services in the core network against malicious use. These mechanisms are developed with the possibility to me enhanced and allow extensions. Interoperability among different users and providers thanks that these security mechanisms have been standardized (Zhang, Zheng and Ma 2008).

The 3G security architecture, shown in Figure 2.23, is composed by five security classes (Zhang, Zheng and Ma 2008):

(I) ***Network access security***: it is related with the security concerning the radio link access. It provides secure connection to the 3G infrastructure for the user and protects it against attacks on the wireless media. This class gives integrity and confidentiality to the user data.

(II) ***Network domain security***: it provides security mechanisms on the wire line networks among different domains. It provides various protocols to improve communications between core networks.

(III) ***User domain security***: ensure access for valid users to the mobile stations. Applications are in charge of performing subscriber, network authentication and key agreement.

(IV) ***Application domain security***: it provides a set of security features to allow secure communication at application level between users and service provid-

ers. Some of the possible services are: entity authentication, message authentication, replay detection, sequence integrity, confidentiality assurance and proof of receipt. An example of this is used in the Wireless Application Protocol (WAP) 2.0, where TLS/SSL is applied to protect the messages at transport layer.

(V) ***Visibility and configurability of security:*** this class allows the users to be informed whether or not a security feature is in operation. Furthermore it provides mechanisms to configure those security features. For example, users can get to know if encryption algorithms are applied to their calls or data and they have the possibility to modify those settings.

Another option to protect the users data across the entire 3G infrastructure is by using *Network-Wide User Data Confidentiality*. In this mode, the data is protected against eavesdropping in all the links of the 3G infrastructure and not only in the wireless link. Network-Wide User Data Confidentiality forces the 3G infrastructure components to exchange encryption keys that will be used by a stream cipher algorithms (Zhang, Zheng and Ma 2008).

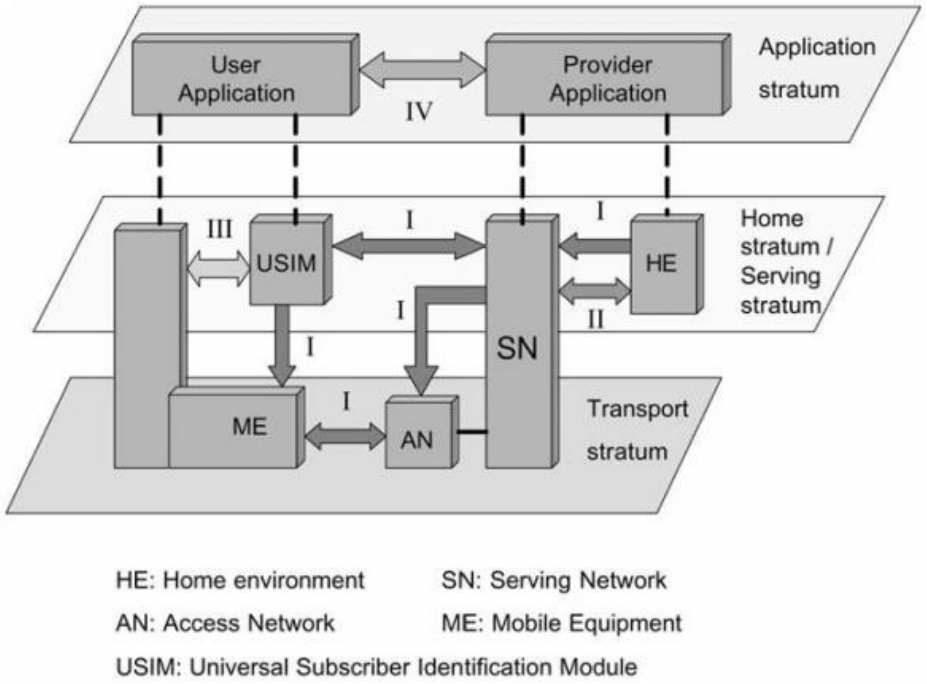


Figure 2.23. *3G security architecture (Chen and Zhang 2004).*

3 SCENARIOS DESCRIPTION

The scenarios that will be described next are real implementations of industrial ambients monitored with WS. They were developed during the SAMIA project. The implementations where done in different automation areas like the process industry, manufacturing plants and the utility sector. This demonstrates that it is possible to monitor centralized (process and manufacturing facilities) and distributed (utility sector) layouts using the same framework.

The monitoring is done at device level, meaning that every device has the capability to provide notifications relevant for monitoring purposes. The monitoring is done in an event-based fashion meaning that every time certain condition is met in the controllers logic an event with relevant data will be sent to the desired recipients. This recipient can be any unit that can provide monitoring services like data: storage, visualization, aggregation or analysis. From now on this component will be referred as the monitoring server.

Event-based monitoring with WS has some differences with polling-based monitoring. The first one is a more efficient use of the network bandwidth since messages are exchanged just when a relevant situation is present instead of polling information constantly waiting for a condition change. Another advantage with event-based monitoring in security terms is that the messages are originated from the device, meaning that messages can be sent to external and untrusted networks like the Internet without the need to have open ports that can be accessed through exterior, which opens a door for outsider attackers. WS events leverages of being built above TCP/IP, therefore messages can be routed to recipients in local or external networks seamlessly. Another difference is that in event-based monitoring the industrial units must know the address of the recipient, compared with polling-based monitoring where the monitoring unit must know the address of each monitored device, which imposes scalability problems.

The most used schema for securing industrial systems is defense-in-depth. Due to the fact that WS are transported over HTTP they can travel very easily through firewalls as outgoing messages, making them very suitable to be used for monitoring purposes in systems protected in a defense-in-depth fashion. Polling-based monitoring would require the proper configuration of the different security shells. The differences between event-based monitoring with WS and polling-based monitoring are summarized in Table 3.1.

Table 3.1. *Event-based Vs Polling-based monitoring.*

	Event-based monitoring with WS	Polling-based monitoring
Bandwidth use	Efficient	Inefficient
Need for open ports to allow external connections	No	Yes
Vertical and horizontal integration	Seamlessly	Configuration modifications required
Information required	Monitoring unit address	Monitored unit addresses
Scalability	Easier	More difficult
Suitable to implement with defense-in-depth	Good	Bad

3.1 Framework

The general framework used in the implementations consisted of industrial devices interfaced and controlled by an industrial controller that implements the DPWS stack. The devices can be monitored either locally or remotely. In case the monitoring is done remotely, a gateway or layer 3 router is in charge of routing the messages through Internet to the monitoring server. The monitoring server concentrates, storages and process the data in order to offer some meaningful information about it through a web-based interface (or other) which can be accessed through the users web browser. This framework is illustrated in Figure 3.1.

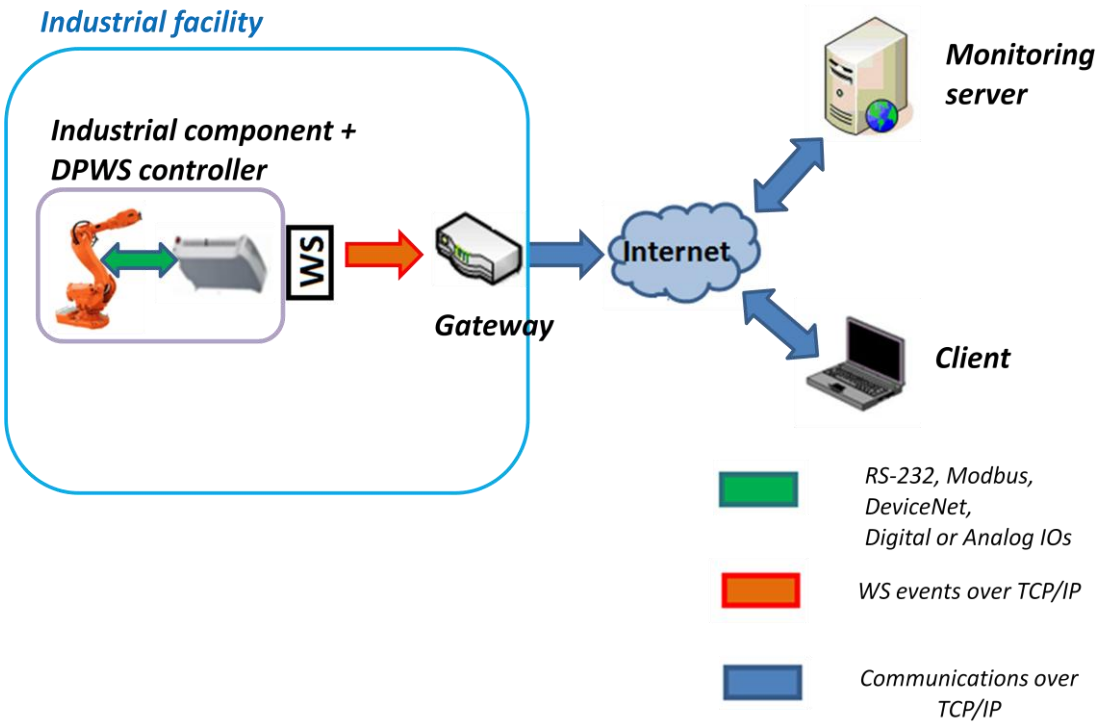


Figure 3.1. *Monitoring of industrial ambients with web services.*

Several industrial components can be connected to the industrial controller, this facilitates the monitoring of digital and analog sensors like presence, flow meters, temperature, pressure or humidity. The state of different actuators can be monitored like valves, motors or even automatic manipulators.

The industrial controller implements the DPWS stack to allow communications in a WS fashion. The control logic is done in Structured Text, and special commands are provided to trigger events. In this way the designer can decide the conditions that will launch and event. The processor of this controller is a 32-bit ARM RISC CPU running at 55MHz and 8MB of flash memory for users programs and data storage (Inico 2010).

The address of the monitoring server, where events will be sent, can be either preprogrammed in the industrial controller or the monitoring server can subscribe to those events with each of the devices. Being the latter the most convenient because it leverages the flexibility and reconfigurability provided when using WS but imposing security concerns since the subscription has to be done from an external network.

It is important to mention that from the monitoring server perspective, the amount of messages might not be a problem, considering that nowadays lot of research and implementation has be done to allow servers to provide service to a huge amount of audience. The same techniques like load balancing or servers farms can be applied here. The critical point that can affect the performance of the system is in the device that implements the DPWS stack. These devices more likely are embedded units with constrained resources for performing computing. Implementing encryption algorithms can affect seriously the performance of the system.

3.2 Centralized Layout

Centralized layouts can be seen in process and manufacturing plants where the industrial equipment is physically located under the same facility. Therefore the devices configuration and maintenance tasks can be done locally. This kind of environments usually has a reliable electrical and communication infrastructure to ensure robustness, thus is possible to have wired access to the Internet.

A key point of centralized systems is that the whole system can be exposed to an external network through one gateway. In this way all the messages which are intended for external recipients go through the gateway. This situation can result in a bottleneck problem if the amount of exchanged messages is too high and the application of authorization policies too complex. The fact of having a single point, where messages go through, rises security concerns as well, since the compromise of the gateway would imply the compromise of different security services for the messages going through it.

The data sensitivity of events which involve just one device can be considered low. For example a conveyor system can notify the monitoring server every time a pallet is transferred. This data might not look relevant for an attacker. In a centralized environment it is likely that the data sensitivity contained in events increases due to the fact that events can be aggregated. For example, lets suppose a workstation with a conveyor system, a material feeder and an automatic manipulator, all of them working in a coordinated manner. The events generated by these units can be aggregated locally and overall equipment efficiency or the production status can be encapsulated in a single event. Thus, the information in this message contains higher sensitivity data. Events aggregation increases the data sensitivity and higher data sensitivity messages are more interesting from the attacker's perspective.

In a centralized system, it is common to have operators with physical access to the network where the automation system is deployed. Also, if the firewalls that separate the plant floor from higher levels within the company are not configured properly, unauthorized personal could gain access to the industrial system. According to a study done by the FBI, 71% of the attacks in corporate networks are caused by insiders(Stephanou 2001). The security attacks originated by insiders can be either active or passive. Active attacks are done with the malicious purpose of damaging the industrial system. Passive attacks are due to human errors, for example an employee could download by accident a Trojan or spyware and infect a computer attached to the industrial network opening backdoors for attacks to the industrial components. At the moment of this writing, the first malware targeting industrial controls systems was released. Stuxnet can attack PLCs in an industrial facility using by modifying code related to the control logic. The malware disseminates through the PLCs by using as vector an infected computer which is attached to the industrial network (Falliere, Murchu and Chien 2010).

3.3 Distributed Layout

In a distributed layout the industrial components are dispersed over a wide geographical area. The utility sector is a common example of this type of environment. The electrical, gas and water infrastructure of a nation belong to this category. Because of the nature of distributed layouts, the industrial components (usually sensors and valves) are located in areas with restricted accessibility, complicating the maintenance of devices and forcing to use remote methods to configure them.

Usually wireless communication technologies are used in this kind of scenarios, because in many cases there is lack of a wired communication infrastructure. The infrastructure that supports the automation system is less robust compared with the centralized systems. For example, in a manufacturing plant, if an energy shortage happens, the key components can be kept running by using a backup power generator, in distributed layouts this redundancy is not possible but still is possible to use less robust redundancy systems like batteries.

The data sensitivity originated in each of the distributed points usually is not high because they report readings of single sensors or valves states. This reduces the wish for an attacker to disclosure the data, but still big compromises can be done to the general system if the messages of the monitoring events are tampered.

Usually personnel are not located in the place where the system is deployed, and the network is limited for the industrial devices, making these environments less susceptible to receive attacks from physical insiders. This does not exclude the possibility of a remote personnel attack by sending wrong configuration parameters to the system devices for example.

The differences between centralized and distributed monitoring layouts are summarized in Table 3.2.

Table 3.2. *Comparisson between main characteristics of centralized and distributed monitoring layouts.*

	Centralized	Distributed
Equipment location	Under the same facility	Geographically distributed
Gateways for external communications	One per facility	One per monitoring unit
Devices management	Local	Remote
Accessibility for maintenance	Easy	Difficult
Communication capabilities	Wired communications (DSL)	Wireless communications (GPRS,3G networks)
Infrastructure robustness	Higher	Lower
Events generation	Higher	Lower
Data sensitivity	Higher	Lower
Possible attackers	Physical insiders + Outsiders	Outsiders

3.4 Scenario A

The scenario A is a flexible manufacturing system and it is a centralized layout case. It is composed by 5 work stations. The workstations are modular and independent from each other. The production is transported through pallets. The pallets can be transferred between the workstations thanks to the conveyor systems that each of them have. The system is shown in Figure 3.2.

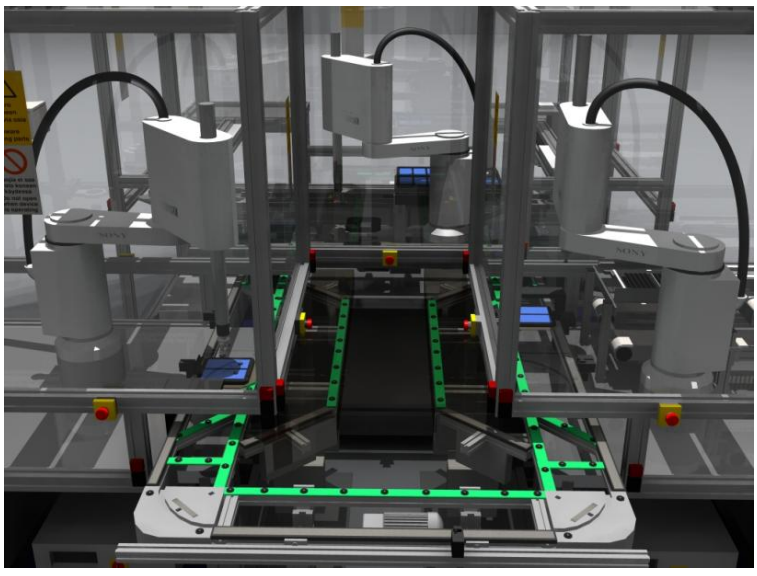


Figure 3.2. *Scenario A, manufacturing system.*

Each of the workstations has:

- 1 SCARA robot
- 1 conveyor system
- 1 material feeder

A total of 15 devices are monitored. Each of these components is controlled/interfaced by the DPWS-enabled industrial controller. All of the controllers are connected through Ethernet to a central switch. A router with Internet connection through DSL is connected to the switch. The routers firewall is configured in such a way that it just allows outgoing connections. This network arrangement can be visualized in Figure 3.3.

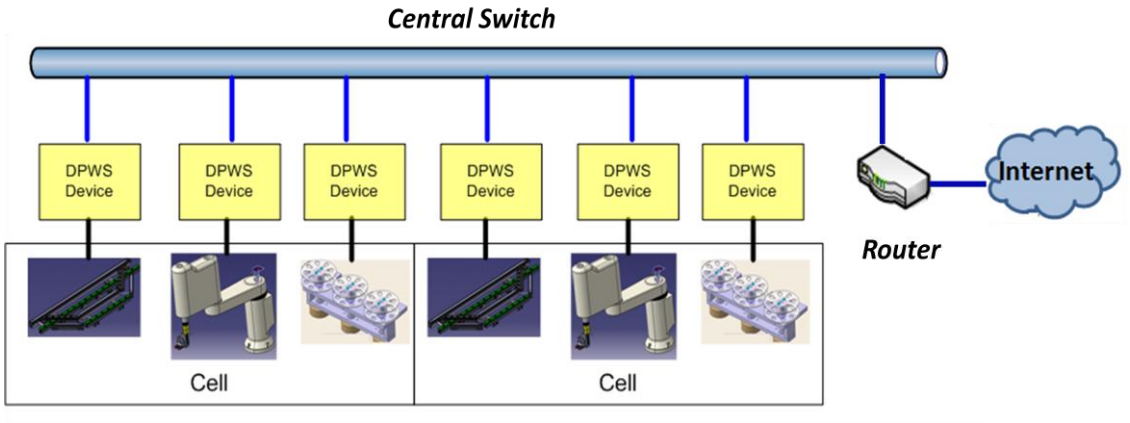


Figure 3.3. *Network arrangement of Scenario A. Modified from (SAMIA 2009).*

When certain conditions are met in the controllers logic, which are relevant for monitoring purposes, an event is generated and intended to the monitoring server. Due to the fact that the publishing of events is done leveraging the WS stack, the events can travel seamlessly to the monitoring server either it is located in a local or in an external network. For this case the monitoring server is located remotely and the Internet is used for transporting the monitoring events.

The DPWS-based controller implements part of the CAMX IPC-2541 standard. This standard defines the communication messages format for electronics manufacturing shop-floor equipment. Under this standard, the equipment has to be in different states like: *Off, Setup, Down, Blocked, Starved, Active and Executing*. When the equipment changes its state, due to the production cycle, the *EquipmentChangeState* event is sent with the proper values to the monitoring server.

The monitoring server storages the events and concentrates the data for each device, in this way the users can see through a web browser:

- Historical trends of the state of each piece of equipment
- The percentage of time the equipment has been in each state
- The Overall Equipment Effectiveness (OEE) of the line

It is common that operators and automation designers have access to the network where the industrial components are connected, for configuring or programming the

devices. This is a vulnerability that can be exploited for compromising the security of the system.

3.5 Scenario B

This scenario shows how the utility sector can be monitored using WS. In this application measurement units were used to read the values of sensors in wastewater facilities. Due to the nature of the utility sector infrastructure, the monitoring units were deployed over a wide are; this can be seen as a distributed system.

The monitoring units are constantly measuring analog signals coming from the sensors. Inside the unit, simple processing is doing on the signal in order to detect the rate of change. When this rate overcomes some predefined threshold an event is send to the monitoring server. This technique proved good enough to save bandwidth and send events just when considerable changes are detected in the sensors measurement. Other conditions that trigger events can also be used, like minimum and maximum values.

Figure 3.4 shows the outside view of a monitoring unit. Each of them consists basically of three components:

- DPWS-enabled controller
- Router + 3G Modem
- Input voltage converter

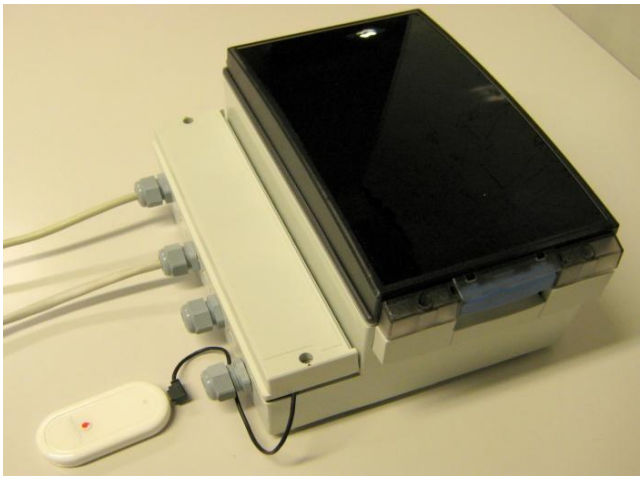


Figure 3.4. *Monitoring unit used in scenario B. Courtesy of THT Control Oy.*

Due to the fact that this units are located in remote places with low or null communication infrastructure, the 3G network is using as communication channel to transport the monitoring events. This is possible due to the fact that 3G networks supports data transfer over the TCP/IP protocol, doing seamless the integration of DPWS-devices with 3G networks.

As it can be seen, the dimensions of the enclosure can be an important constraint if more elements need to be added to improve the robustness of the system. Therefore it is also complicated to add extra components to improve the security of the system.

Even though the events generated by each of the nodes can be low, the amount of events received by the monitoring server can be high, depending on the application. The data gathered and summarized by the remote monitoring server which runs on ClearScada can be accessed through WebX clients as shown in Figure 3.5.

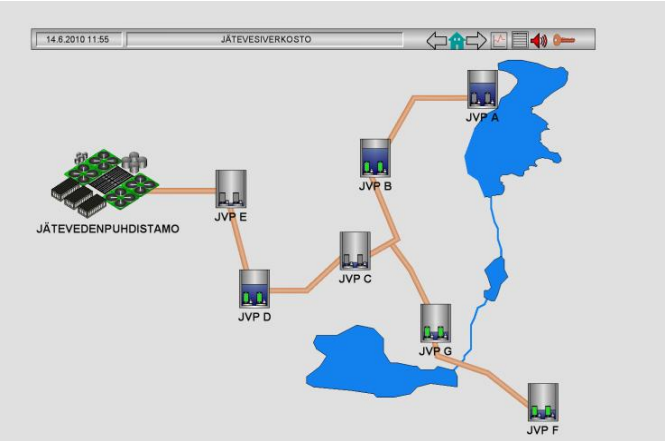


Figure 3.5. *User interface used in Scenario B. Courtesy of THT Control Oy.*

3.6 Monitoring Service with Multiple Intermediaries

Usually a monitoring service is composed by two endpoints: The industrial facility that generates the messages and the monitoring server that storage and processes them. These two endpoints can belong to the same organization, or the monitoring server can belong to a trusted third-party.

The use of WSs, as the core communication technology, opens the possibility to involve multiple intermediaries in the monitoring service. Each of them providing a specialized service in order to improve the overall quality of the monitoring application. Monitoring services could be offered in the next categories:

- Data aggregation
- Visualization
- Data mining and processing
- Storage

In order to illustrate this, Figure 3.6 exemplifies a scenario where the industrial ambient is producing monitoring events which have to go through multiple recipients. Firstly the message goes to a data aggregation and visualization provider, then the message are forwarded to the data storage service.

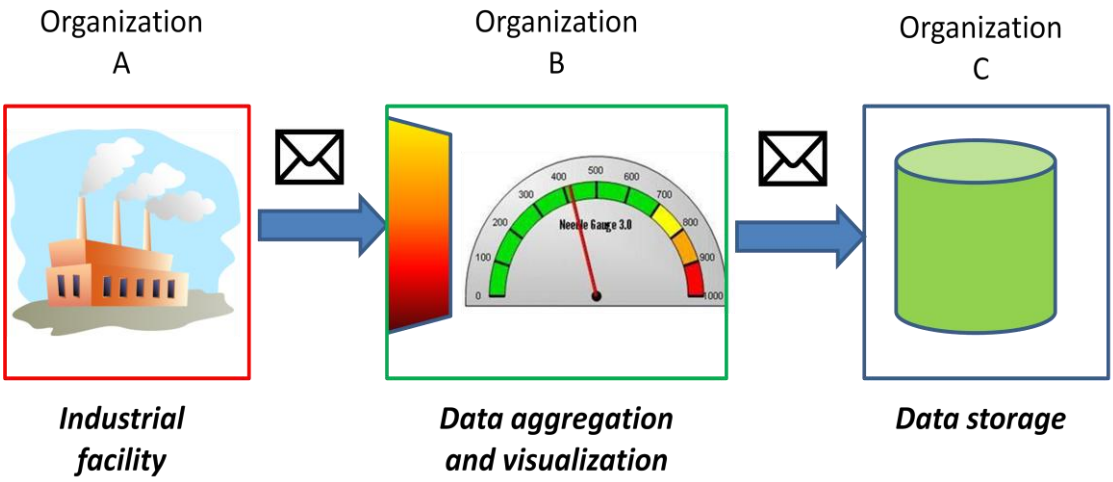


Figure 3.6. *Industrial monitoring with multiple intermediaries.*

The interoperability between the service intermediaries is empowered by using WSs which can easily allow the communication between different platforms. WSs also provide mechanisms to allow the transfer of messages among multiple intermediaries. Each of these intermediaries can attach pieces of data to the message if needed. Another advantage of using WSs in this scenario, is that using security at SOAP messages allow to have different security characteristics for different recipients.

The possibility to have different security contexts allow to safely transfer a message through different entities preserving some security services, for each of the message recipients. For example:

- Integrity of the message can be warranted so Organization B and C cannot tamper its content.
- Confidentiality of the message can be provided in such a way that only organization B can access some sensible data fields. Organization C will storage the message without having access to those fields.

This situation, where multiple intermediaries participate in the monitoring service, is introduced to highlight the importance of including security in the monitoring events. By providing security services at message level, multiple participants can be used in the monitoring service even though they belong to different domains of trust.

Including QoS in this chain of services is very important to allow the different parties to agree about terms and conditions of the service they provide. For example, Organization B might demand that the monitoring events coming from the industrial facility include some local processing at device level. So QoS is a step forward for automatic agreement of conditions between service consumers and service providers.

4 SECURITY ASSESSMENT

The main purpose of this chapter is to assess and analyze security issues that arise when monitoring industrial ambients with WSs. In order to start the analysis, the functional and security requirements of the application are identified. Due to the fact that the system is monitored with DPWS-enabled devices, which are resource constrained, a discussion is made with the aim to understand the feasibility to implement security on them. The scope and extensibility of the DPWS standard is also reviewed.

Based on the previous analyses and discussions, different security mechanisms are assessed, possible threats are identified. Proposing a unique secure framework that will cover all the scenarios is unfeasible, instead of that couple of decision diagrams are presented in order to select the architectural components and the security protocols depending on the monitoring application characteristics. These guidelines for creating the framework make use of the security mechanisms assessment and threats identification analyzes.

4.1 System Requirements

In order to securely monitor an industrial environment that employs WSs, the system must accomplish some requirements. The requirements are divided in two categories, functional and security requirements. Functional requirements are those which need to be fulfilled so that the system can accomplish its primary objective, the remote monitoring of an industrial environment. The security requirements are those security services that should be warranted so that the system can be considered secure in cyber security terms.

4.1.1 Functional Requirements

The requirements identified in this section were obtained based on the reviewed literature and the scenarios described in the previous chapter. The requirements and their values might vary depending on the application.

Monitoring messages type

Remote monitoring with WSs is suitable for *soft real time* and *non real time* messages. Using Internet as the communication network implies considerable delays in the data transmissions and the verbosity of XML-based messages contained in WSs demands valuable computing time to parse them. These conditions leave out of the scope the monitoring of data that has *hard real time* requirements with WSs.

The monitoring messages can be classified in different types with different priorities. In case of messages queues in the monitoring server, their priority should be used to decide their processing order. Table 4.1 enumerates the different types of messages that the monitoring server can receive, their priority(where 1 is the highest) and the tolerated delays[2] for the message arrival considering the transmission and processing time.

Table 4.1. *Monitoring messages type, priorities and tolerated delays.*

Monitoring messages type	Priority	Tolerated delay (s)
Alarm	1	1-2
Errors	1	1-2
Warning	2	2-3
Equipment Statue Change / Signal Rate Change	2	2-3
Info	3	3-4

Monitoring layouts

The monitoring framework should be able to work on centralized and distributed layouts. The idea is to preserve the logical components (DPWS-enabled controller) and communication platform (WSs) and just change components which are dependent of the available infrastructure (wired and wireless interfaces to connect to Internet). With this is possible to remotely monitor industrial environments like the process and manufacturing plants and facilities in the utility sector with the same framework.

External subscription to devices events

In order to exploit the advantages of using WSs at device level, the monitoring server should subscribe to the events generated by the industrial devices. This improves the flexibility and reconfigurability of the system because any monitoring entity can subscribe and consume the data produced by the industrial units. In this way, the industrial facility is a service provider of data.

Unfortunately the DPWS considers that the service discovery and events subscription is done in a local network scope. This means that an external monitoring server cannot subscribe to the industrial units. Fortunately this issue can be bypassed by using a DPWS proxy as documented in (Muller, et al. 2010). For this application the interactions between the components would be as follow: The DPWS proxy registers all discovery services in the industrial network. The monitoring server (located in an external network) requests the available WSs to the DPWS proxy through a secure connection like TLS/SSL. The monitoring server chooses those devices that will be monitored and

[2] The tolerated time delays vary according to the application. The proposal of these delays is according to the tested scenarios.

uses the DPWS proxy to subscribe them using as the end point address that one of the monitoring server. Every time a monitoring event is generated at device level, the message is sent to the monitoring server. These interactions are represented in Figure 4.1.

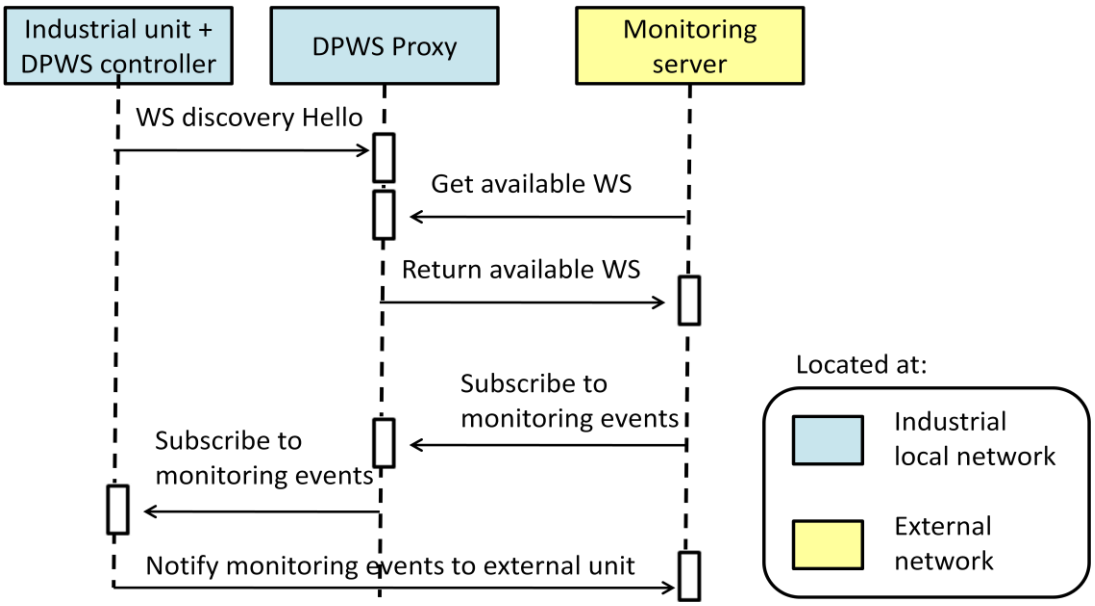


Figure 4.1. *Subscribing to DPWS controllers from external networks.*

4.1.2 Security Requirements

The security requirements listed in this section enable the secure transmission of monitoring data from an industrial facility to a remote server using Internet as the communication network. They were chosen with the aim of allowing system scalability and flexibility and having in mind the performance impact of using encryption algorithms in embedded devices like DPWS controllers.

Relevance of security services

When monitoring an industrial system it is a primordial requirement that the data is protected against any alteration during its transmission. If the monitoring data is tampered with false values the system can suffer serious damages. For example, if a monitoring centre is checking the pressure of a gas tank and the value of the measurement is modified by an attacker, there is the possibility to have dangerous pressure levels but the monitoring centre might perceive that the tank pressure is at safety values. That is why data integrity is necessary in industrial communication systems for control and monitoring purposes.

In this application the monitoring events are generated at device level. This implies that the data sensitivity of the messages is rather low. Usually the monitoring events will have values concerning the readings of a sensor or the state of some piece of equipment. Very likely this is out of the interest for an attacker, unless the reading represents some critical piece of information that can be used for further malicious purposes. If data or event aggregation is done within the industrial environment then the data sensitivity increases as it was discussed in the previous chapter. In case protection

against the disclosure of the monitoring events data during the transmission is wished, confidentiality services must be implemented.

From the monitoring server perspective, it is important to authenticate the sources of the events, in order to process and analyze the data properly. The authentication can be done through simple credentials like username/password or more sophisticated ones like SAML tokens could be used. If the industrial devices allow remote subscriptions to their events, it is necessary that the device authenticate the service requester. Username/password or other tokens can be used for this.

The monitoring events does not contain commands or instructions, they contain passive data in the sense that they do not request any kind of control over resources. In this way, the authorization service is not required at the monitoring service. Similar to authentication, if the industrial devices allow remote subscription to their events, it might be required that they can authorize which granularity or type of events the service requester can subscribe. In this case the DPWS devices must implement authorization mechanisms.

Non-repudiation service is useful to demonstrate that transactions did occur. In a monitoring service that involves third parties this can be useful to resolve disputes among the parties. Having this property can be useful when doing audits of the system transactions. Non-repudiation can be used as well as a mean to provide proof of integrity and origin of data.

Table 4.2 summarizes the relevance of the security services for the purpose of monitoring industrial ambients. The table includes their priority, the conditions in which they should be applied, the implementation side (monitoring server / DPWS device) and an estimate of their computational demand[3].

[3] The computational demand estimation was done based on the reviewed literature.

Table 4.2. *Security services relevance*

Security service	Priority	Apply in case of	Implemented at	Computational demand
Integrity	1	• Always	DPWS device	Low
Authentication	1	• Server side needs to authenticate source • DPWS controllers offer monitoring services to external network entities	Monitoring server/DPWS device	Low/Medium
Confidentiality	2	• Event/data aggregation is done in the industrial environment.	DPWS device	Heavy
Authorization	3	• DPWS controllers offer monitoring services to external network entities	DPWS device	Low/Medium
Non-repudiation	4	• Interaction with third parties • Audit ability required	Monitoring server/DPWS device	Heavy

Point-to-point Vs End-to-end security

When using point-to-point security, a secure transmission is warranted between two points by establishing a secure channel. This is done by applying security at the transport layer by using some mechanism like TLS/SSL or IPSec. In this way confidentiality, integrity and authentication are provided at the transport level.

End-to-end security allows securing messages even though they have to go through different intermediaries. Under this schema different participants can have different security privileges over the message, for example some participants might be able to modify the message data while others not or some participants might be able to read some fields while the same fields will be unreadable for other participants. This is possible by applying security mechanisms at message level. These security approaches are represented in Figure 4.2.

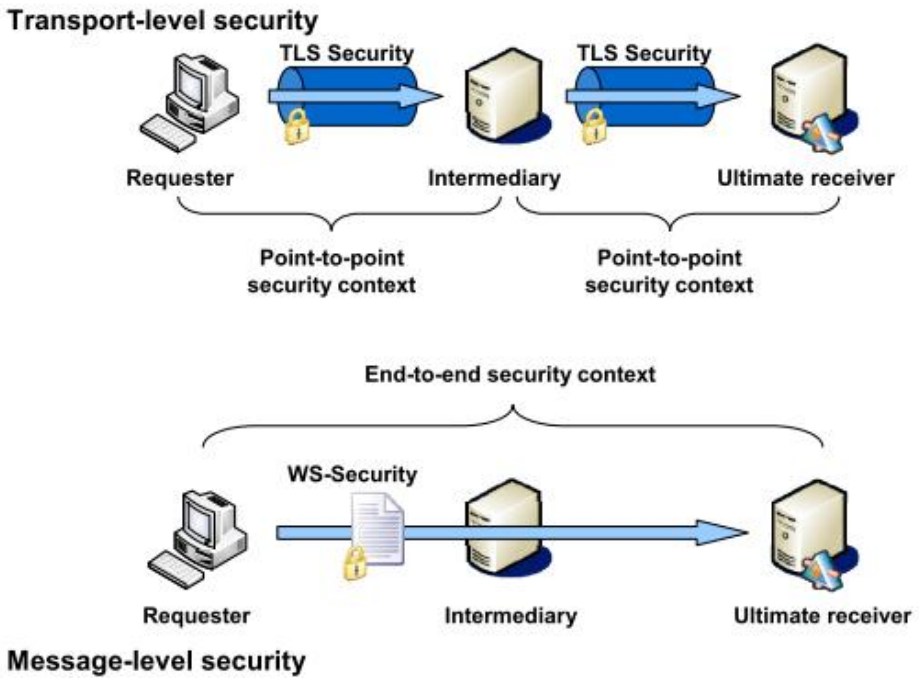


Figure 4.2. *Point-to-point and end-to-end security (Martinez, et al. 2008).*

If the monitoring process is done just between two endpoints point-to-point security might be enough but if interoperability, scalability and reconfigurability must be addressed, end-to-end security offers a better platform. Providing security at message level is more flexible and different protection methods can be applied from a pool of security mechanisms. So the security designer can apply different levels of security depending on how critical the application is and the security services that must be provided.

The application of this thesis uses WSs as core of its communication technology, therefore it is possible to use WS security to provide security services at the message level and take advantage of the end-to-end security privileges.

Defense-in-depth

Defense-in-depth is the most accepted and used security schema in industrial systems. As already described, defense-in-depth is done by implementing different layers of secure mechanisms around the protected asset. It allows the security designers to decide what kind of security mechanisms they want to implement and their strength level. A very common component used in this schema is the use of firewalls to separate sensitive networks from the rest of the organization networks.

In an event-based monitoring process just outgoing messages are required, no bilateral communication is required. Since the messages are encapsulated as WSs events and transported over HTTP, they can travel very easily through firewalls as outgoing traffic. Security at message level does not affect the routing of the messages and high layers components can analyze the traffic, which is useful when using high level IDSs. Another advantage of using security at message level, is that if the industrial components layout changes, or the defense-in-depth shells are modified, no major modifications are required in the firewalls configuration.

In contrast, when security is implemented at transport level, some tunneling protocols are difficult to be routed; this has been documented for IPSec. Security at transport level implies that the data payload is encrypted, therefore a high layer security component cannot be aware of what kind of information the system is sending to the exterior.

Reuse of existing components

An important design aspect is to reuse as much as possible the existing components in the monitoring infrastructure to improve the security. In the centralized layouts a gateway is always used to allow the DPWS controllers to send their events through the Internet. This gateway is usually has an embedded firewall. This firewall should be used as security mechanism. A gateway that implements natively VPN connections can be used to reduce the computing load in the DPWS devices when performing encryption algorithms. It is recommended that the gateway is of the industrial type to warranty robustness.

In the case of distributed layouts, 3G networks are used to provide wireless communication to the system. As already reviewed, the 3G network offers secure services for data transmission. By using these services, data can be protected without forcing the DPWS devices to perform encryption algorithms. Unfortunately 3G secure services are very dependant of the mobile operator companies and some might not offer them. It has to be considered that by using tunneling connections in the gateway or 3G secure services the advantages of using message level security can be lost because encryption is applied at transport layers.

Local Vs Federated Authentication/Authorization

If the case where the monitoring server subscribes to the DPWS devices events, it is necessary to use some authentication/authorization mechanism at the device side. Since the DPWS device acts as a service provider it must decide whether to accept the subscription request or not.

The authentication mechanism can be implemented easily if the device is provided with a list of valid clients. Another list with the events that each client is allowed to subscribe can be used as an authorization mechanism. Both of them are example of local authentication and authorization services.

Implementing local authentication and authorization like indicated previously has the advantage of being simple and computationally simple for the DPWS devices, but it has two disadvantages: The authentication and authorization lists should be maintained in all the devices, limiting a lot the scalability of the system. For example new clients cannot be added without updating all the devices lists first. The second disadvantage is that the device can be very easily fooled if an impostor presents a fake credential which is registered as a valid consumer. This can be solved by using digital signatures but this can be an intensive resource consuming task for the DPWS device.

Federated authentication and authorization offers a more suitable platform for scalability and interoperability. A federated scheme is used to decouple the authentication

and authorization verification from the device functionality. This is done by using a trusted server which is in charge of updating a list of valid clients and the events they are allowed to subscribe. Once the DPWS device receives a request with the credentials or tokens of the client, it forwards the request to the authentication and authorization which makes the validation process. Then it responds to the DPWS device whether to grant or not access.

Federated authentication and authorization is desirable for this thesis, having in mind that they allow further reconfiguration of the system and interoperability among different participants. The credentials management is done at a single point facilitating this task. A remote authentication and authorization server is desired so that it can be used in both layouts, centralized and distributed.

4.2 Feasibility to Implement Security in DPWS Devices

Having WSs at device level promises the interoperability of these components with higher level applications and platforms. Technically, a WS hosted in an embedded device can be consumed, published/subscribed and orchestrated in the same fashion that a WS hosted in a dedicated server. The main difference comes in terms of performance; due to the fact that embedded devices have slower processors and smaller memory capabilities to execute processes when compared with standard PCs. Knowing the limitations and capabilities of DPWS-based devices is useful to analyze the feasibility to implement security mechanisms at device level.

Another key point to have into consideration is the scope of the DPWS standard. It is important to analyze how flexible this specification is, so that it can be adapted to the requirements of the application and the resource constrains of the devices. The adaptability of the standard to incorporate secure mechanisms, with different levels of complexity and computational demands, is another indicator of the feasibility to implement security in DPWS-based devices.

4.2.1 DPWS Devices Constraints and Capabilities

Cryptographic algorithms can be very resource demanding even for computers with standard computational power, therefore a wrong selection of the cryptographic suite can decrease considerably the performance of the embedded device. The processors speed and the amount of RAM memory vary depending of the device, but considering as reference the current DPWS-based controller they can be around 8MB of memory for data execution and storage and running at 55MHz.

With these resources, a DPWS-based controller must be able to work as a control unit and as web server simultaneously. As a control unit it should storage and run the control/monitoring logic of the process/device it interfaces. As a web server, the controller must decrypt and parse the messages it receives. The output messages should be encrypted, wrapped in the SOAP format and sent to the proper recipient. The general architecture of a DPWS-based controller is shown in Figure 4.3.

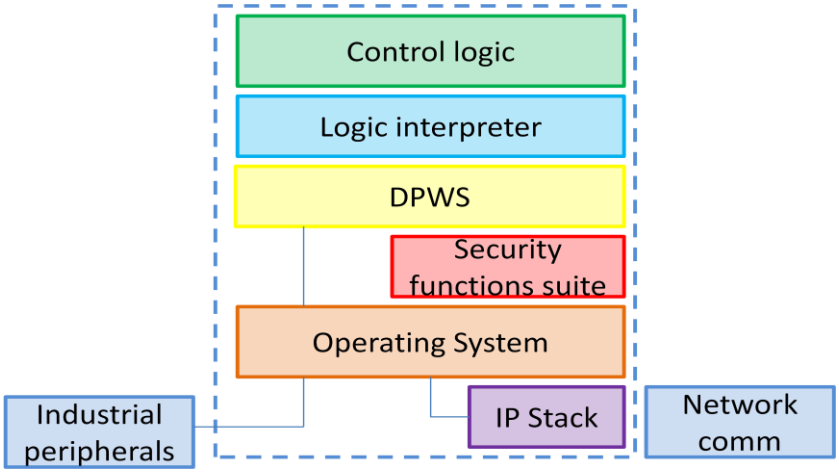


Figure 4.3. *DPWS-based controller with security mechanisms.*

The average size of a monitoring event is around 800 bytes. This is due to the fact that XML is a verbosity format. This size of message is comparable to the size of messages (1KB) used in the test realized by (Delamer and Martinez Lastra, Information Security for Reconfigurable Manufacturing Systems using Networked Embedded Controllers 2006), described in chapter II, where different cryptographic functions where tested in a networked industrial controller with similar computational capabilities (100MHz, 2MB RAM) to this application controller. Thus, their results can be applicable to this case.

In their test, depending on the selected cryptographic function, the encryption/decryption total time of a 1KB message can range from 132 ms (using AES-128 bit) to 737 ms (using Rijindael 2.0 – 128 bit). The creation and encryption of a hash of the message can range from 20 ms (using MD5 – 128 bit) to 9 401 ms (using PSS Signature Generate 1024 bit).

The results obtained in (Gupta, et al. 2005) proof how embedded devices with even smaller computational capabilities (around 10 MHz, 50 KB RAM), can implement a secure web server using SSL with public key encryption algorithms. With this is possible to provide proof of origin, integrity and authenticity to the messages. Their results for four different types of motes using different cryptographic functions are shown in Figure 4.4.

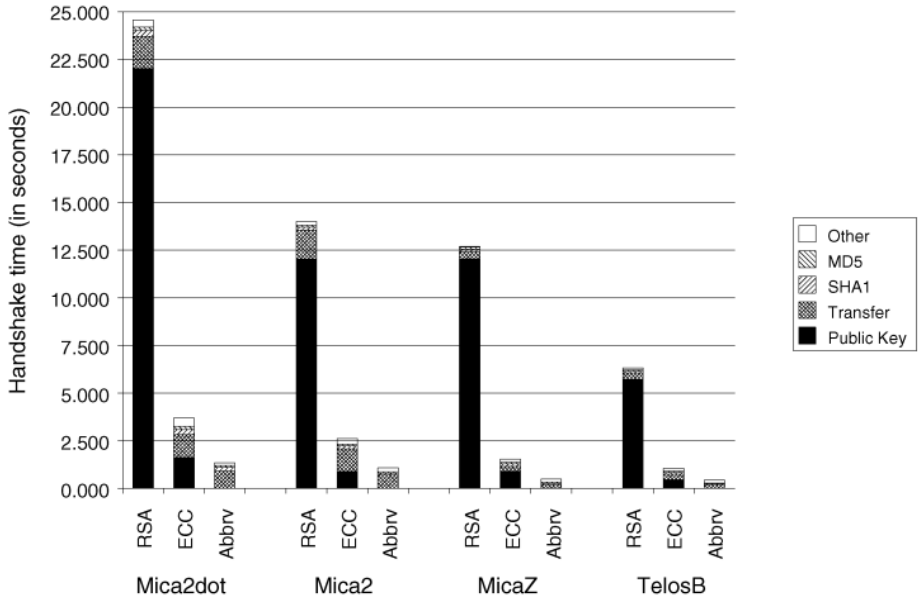


Figure 4.4. *SSL performance in resouce-constrained devices (Gupta, et al. 2005).*

As we can see in these tests, again the selection of the cryptographic function can have a big impact in the performance of system. In this study it is demonstrated that Elliptic Curve Cryptography (ECC) is suitable for embedded devices. ECC offers equivalent security level as using RSA algorithms, but with significant smaller key sizes (Gupta, et al. 2005).

Symmetric and Asymmetric algorithms

Even though is well known that symmetric algorithms are substantially less demanding than asymmetric algorithms, the latter ones offer well known capabilities for scalability. Asymmetric algorithms allow the implementation of public key infrastructures used for authentication/authorization through a third party. This is desired for the secure framework of this thesis. As described in the previous point, embedded devices can be used with asymmetric methods. For industrial DPWS-based controllers, asymmetric encryption should be used just for establishing a secret key that later will be used in symmetric encryption.

Preemptive scheduling

The industrial controllers work in a scan cycle manner. Usually for manufacturing plants, the control can be accomplished with scan cycles in the range of hundreds to tens of milliseconds. After discussing the time it takes to an embedded controller to process a message with cryptographic functions, it is possible to notice that the application of these algorithms cannot be executed within one scan cycle.

In order to process the message with cryptographic functions, the operative system of the controller must implement some preemptive scheduling, in such a way that the cryptographic task can be interrupted and resumed after the controller computes the control logic. In this way the controller can finish the cryptographic computation after

some finite number of scan cycles without affecting the control of the system. It is necessary to remember that the monitoring information should not have real time requirements.

Amount of subscribers

The amount of subscribers that receive the monitoring events of the DPWS-based controller is a variable that can affect the performance of the unit. Every time an event is generated, wrapping and cryptographic functions should be invoked. If the message is sent to different recipients and encrypted with different secret keys, the processing load on the controller can be too high.

Despite of that, every time a message is sent, the TCP protocols expects an acknowledge reply. This can slow down the device if it waits for different acknowledge replies. Therefore it is recommended that a monitoring event is sent to a higher resource computer that can forward or broadcast the message.

4.2.2 Security Scope and Extensibility of the DPWS Stack

In the security section, the DPWS standard recommends a minimal set of technologies to provide: integrity and confidentiality of the exchanged messages and authentication between devices.

The DPWS proposes the protection of integrity and authentication of WS-Discovery messages by applying message-level signatures at the UDP packets. Once the discovery process has been done, the standard recommends the use of secure point-to-point communications by using TLS/SSL. This ensures the authentication of participants plus provides integrity and confidentiality of the exchanged messages since the data is transported over HTTPS.

After reviewing the DPWS standard, the next is a list of considerations which are out of the scope:

- End-to-end communications are not covered.
- Authorization and Non-repudiation services are not covered.
- UDP WS-Discovery messages are not treated as confidential, so sniffing of the network is possible.
- The credentials management, distribution and validation mechanisms are out of the scope.
- Devices discovery is not meant to be done from external networks.

The standard is conservative in the sense that it prioritize scenarios are deployed under local networks with relative low number of participants. In this sense, the system scalability is not a priority but rather its functionality and performance.

Even though the security recommendations of this standard are limited, it gives great flexibility for the designers to implement other mechanisms and security profiles. If the security profile is meant to be extended, it is responsibility of the designer to war-

ranty the composeability and interoperability of the mechanisms with the DPWS specification.

The device can make use of the WSDL file or WS-Policy messages to inform clients about the security requirements. For example, they can be used to communicate which cryptographic methods are used. Another key point is that a device can host secured, non secured or both services. This is especially useful, for this thesis application, because monitoring events within the local network or a trusted party can be sent without security services while events designated to external networks or untrusted parties can be secured. This granularity provides flexibility.

4.3 Security Mechanisms Assessment

This section discusses how the security mechanisms, presented in the theoretical background, can be used to secure the communications when monitoring industrial ambients with WSs. In the context of this application, their advantages and disadvantages are highlighted.

4.3.1 Miscellaneous Mechanisms

In this section, security mechanisms belonging to different security categories are discussed. These security mechanisms include security at transport layer, tunneling protocols, security appliances and preventive systems among others. In this way, the mechanisms presented next act on different layers of the OSI model.

TLS/SSL for transport layer security

If the TLS/SSL mechanism is used between the DPWS device and the monitoring server, it is possible offer integrity and confidentiality on the exchanged messages and server authentication. It is possible to be used in embedded devices if a light cryptographic suite is chosen. By using this mechanism the SOAP messages of the WSs will be transported in the so called HTTPS.

In an scenario where the DPWS device allows subscriptions from external networks, the monitoring unit can authenticate the identity of the DPWS device. In this case the device must provide a certificate with its public key and a trusted Certificate Authority address.

TLS/SSL is recommended by the DPWS standard to offer point to point security. This mechanism can be used in centralized and distributed layouts, but in it can become complex for the latter case, if the amount of participants is big and a key management system is not properly implemented.

***Advantages*:** This protocol is widely used in applications which require critical transactions. It is flexible and allows the participants to select the cryptographic suite. When it is used with a certificate authority, the system can be highly scalable.

***Disadvantages*:** TLS/SSL is meant to enable point-to-point secure communications, so it is not suitable to be used in a scenario with multiple intermediaries. Because it uses

transport level security, it limits the flexibility of the system, for example, message forwarding is constrained. If an IDS is used to analyze the content of the messages, it can have conflicts with this mechanism.

Tunneling protocols

Tunneling protocols can be implemented between the gateway that interface the industrial components and the monitoring server. They can provide integrity, confidentiality and authentication services.

This mechanism is suitable for centralized layouts because a gateway with VPN capabilities can be installed to interface the industrial infrastructure. In distributed layouts this is not practical since this means having a VPN for each monitored device.

***Advantages*:** Once the tunnel has been established, elements from different networks can interact with each other in the assumption that they belong to the same network. This mechanism has proved to be secure in high data sensitivity applications and it is widely used in corporations to allow remote access to their resources.

***Disadvantages*:** Difficult to accomplish key management tasks. It can have problems with firewalls and IDS since the data is encrypted at middle layers of the OSI model. Some tunneling protocols present problems when the traffic needs to be routed. Not suitable for scenarios with multiple participants. The communication is supposed for two trusted parties, reducing the reconfigurability and flexibility of the system to be integrated with dynamic participants. Usually this mechanism does not provide fine granularity authorization policies.

Intrusion Detection System

In a monitoring application, the IDS can be positioned in the industrial network to analyze the traffic of messages with the aim to find possible attacks. For example, DoS attacks can be detected if a device is receiving an abnormal high number of requests or subscriptions for events. An IDS can be used to detect unauthorized access to resources. For example, if a client is attempting to subscribe to events which are meant for other purposes rather than monitoring, the IDS can evaluate this requests for subscriptions in the system policies and find possible violations.

This security mechanism is suitable for centralized layouts, because extra components can be added to the industrial network easily. In distributed layouts this mechanism is not suitable since the infrastructure cannot be modified so easily and an IDS would analyze the activity of just one device.

Advantages: Policies violations and malicious activities can be detected without modifications to the industrial network, allowing the reconfigurability of the system.

Disadvantages: If not configured properly, the false-alarm rate can be too high. Signatures of attacks are needed to feed the IDS; there is very little common knowledge of the profile of cyber attacks on industrial systems. It is problematic to validate policies violations or unauthorized access to resources if messages are encrypted at middle layers of the OSI model.

Security appliances

Layer 3 firewalls are used to filter traffic based on IP addresses. The gateways used for interfacing the monitoring devices to the Internet have already implemented this type of firewalls, so they can be used without adding external components to the system. This type of firewalls can be used in the organization to separate the industrial network from the management and personnel network. Depending on their configuration they can be used to avoid DoS attacks.

Layer 2 firewalls can be used to establish security policies at MAC level. They can be used in an industrial network to avoid IP spoofing attacks. Layer 7 firewalls filter the messages by analyzing the SOAP messages, so if the filter policies are specified correctly, they offer a good solution as an authorization mechanism.

Layer 3 firewalls must be used in both, centralized and distributed layouts, because it is part of the current infrastructure. Layer 2 are recommended just for centralized layouts because they are susceptible to IP spoofing attacks. Layer 7 firewalls are complex units and therefore it is not practical to use them just to interface one piece of equipment; they are also recommended to be used just in centralized layouts.

***Advantages*:** No need to modify the source code or functionality in the DPWS devices. WS messages can flow easily through firewalls as outgoing traffic (monitoring events). Layer 3 firewalls are already implemented in the gateways used for monitoring centralized and distributed layouts. A modification in the network layout implies small or null modifications to the firewalls configuration for a monitoring application. It can be easily used to implement layers in a defense-in-depth schema. Firewall appliances are widely used in cyber security, therefore its product range is very huge; it is possible to find commercial industrial firewalls. The computational work is done on the firewall appliance and not in the DPWS devices.

***Disadvantages*:** Especially layer 2 and layer 7 firewalls are potential bottleneck points, because layer 2 firewalls have to filter an extensive amount of messages, while layer 7 firewalls have to do time consuming tasks like XML parsing. Another disadvantage for firewalls is that their configuration has to be set according to the necessities of the application; a security hole in their configuration can compromise the network behind them. Their configuration management can become messy if the amount of firewalls is too high. If encryption is done at middle layers of the OSI model, layer 7 firewalls might encounter problems to apply filter policies.

Secure services in 3G mobile networks

If the monitoring application uses 3G as wireless mean for transporting the information, then it is possible to leverage some of the security services provided by the 3G network infrastructure. By using the class I (Network Access Security) of the 3G security classes, the data coming from the DPWS devices is protected with integrity and confidentiality services in the its wireless link. This service is provided by default when using 3G networks.

The 3G security class IV (Application Domain Security) can be used to provide services at application level between users and service providers. This implies that the data transferred between the DPWS devices and the monitoring server would get some security service provided by the mobile operator. Some of these services are: entity authentication, message authentication, replay detection, sequence integrity, confidentiality assurance and proof of receipt. This service requires that both end points support the security services provided by the mobile operator, thus reducing the reconfigurability capabilities of the system.

If the data coming from the industrial environment is high sensitivity data (which might not be the case as already described in the previous chapter), then the Network-Wide User Data Confidentiality mode can be used so that the monitoring data cannot be disclosure in any of the links within the 3G infrastructure.

The use of this services makes sense just when using 3G networks and therefore its applicable just for monitoring systems with a distributed layout.

***Advantages*:** By using the services offered by a 3G mobile operator, the existing infrastructure is reused which implies no adding extra components. By default the wireless link is protected. Security processes are done on the 3G infrastructure components. Encrypted data at application level can be managed by 3G infrastructure; in order to increase security and avoid using some of the services provided by the mobile operator.

***Disadvantages*:** Security services of class IV are rarely available in commercial use. Using class IV services imply that security is provided in a point-to-point schema disabling the possibility to integrate multiple participants in the monitoring service. The system reconfigurability is severely affected, since the new participants in the application will have to support and agree the terms established by the mobile operator. Management of the security settings can be costly.

SAML and XACML

SAML and XACML can be used together to provide a federated authentication and authorization mechanism for the validating the identities and rights of the participants in the monitoring system. By using SAML is possible to do assertions about the participants respect to their identities and attributes. It is a flexible mean to distribute security-related information of the participants which can be used for different purposes.

For example SAML can be used in a federated authentication schema during the subscription process of a monitoring server to the monitoring events of an industrial device. First the monitoring server attempts to subscribe to the events generated by the industrial component, the subscription is done through a DPWS gateway. Then the industrial component forwards the credentials of the monitoring server to a SAML identity provider (IdP). The IdP has a repository of credentials and attributes of valid monitoring servers (or other participants). The IdP generates a SAML message which contains information about the authentication, like period of time in which the credential is valid and attributes which contain some details of the authenticated party like it monitoring category for example, maintenance or production monitoring. It is important to

mention that the authentication process can be delegated to the DPWS gateway in order to reduce computational processes in the DPWS devices.

XACML is used to describe the access control policies to gain access on network resources. It specifies an interaction model with different components that enable the authorization process. It also defines a request/response message language between those components. Authorization decisions are based on the authentication and attributes of the requester, in this case, the monitoring server. SAML by itself provides a mechanism to represent these assertions and attributes, XACML is used to evaluate them and decide if it grants access to that resource or not.

The DPWS devices can contain XACML policies that define how they should process the SAML assertions of the monitoring server. Depending on its assertions, the DPWS device can authorize subscription to certain type of monitoring events. For example, a monitoring server with production monitoring attributes might not be able to subscribe to maintenance monitoring events.

It is important to mention that the flexibility offered by XACML allows to move the decision point out of the target device by using Policy Decision Points which can be deployed on the device or in an external server, being the last one the most suitable for this application which uses embedded devices.

These technologies offer a good platform for reconfigurability and scalability. In case the interaction can be handled by the DPWS devices it can be used in centralized and distributed layouts.

Advantages: These technologies are platform independent. It reduces the management effort of maintaining credentials by using a single point for credentials repository and validation. These technologies are flexible and can be used combined with other standards. They allow the reconfigurability and scalability of the system. Heavy computational process are done on authentication and authorization servers and not in the DPWS devices.

Disadvantages: They parties who exchange SAML assertions must know prior the semantics of the exchanged data; this can be solved by using an ontology. The participants which need to be validated must be registered first in the IdP.

4.3.2 Web Services Security Suite

The WSs security suite is composed by different specifications which facilitate the interoperability and exchange of secure messages. The security mechanisms provided by WS-Security are leveraged by other specifications like: WS-Policy, WS-Trust, WS-Pricacy, WS-SecureConversation, WS-Federation and WS-Authorization. One of the biggest promises of using WSs is that specifications can be added to an application in order to address certain needs. Therefore, the previous security specifications can be used according to the security requirements of the monitoring application.

Once the desired specifications from the WS Security suite have been selected, they and their dependencies must be implemented within the DPWS stack of the industrial controller, as shown in Figure 4.5. Then the functional WSs specifications, those that

are essential for the functionality of the monitoring application, like: WS-Discovery, WS-Eventing, WS-Addressing and WS-Transfer can make use of the WS Security suite functionality.

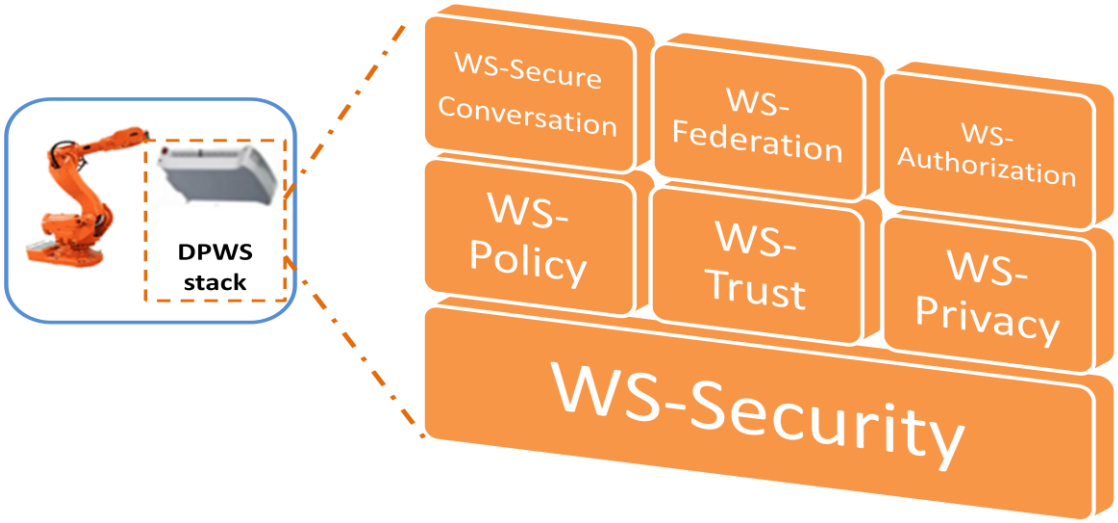


Figure 4.5. *WS Security suite within the DPWS stack of an industrial controller.*

WS-Security

WS-Security it used to provide security to SOAP messages on WSs. For this application it can provide integrity, confidentiality, authentication and Non-repudiation to the monitoring messages. One of the main objectives of this specification is to allow the use of security tokens like: SAML, Kerberos and X.509 tokens, being the last one the recommended by the DPWS standard. These tokens are used mainly for authentication, authorization and agreement of encryption key purposes. WS-Security relies heavily in the use of XML Encryption and XML Signature.

It is important to mention that other secure WS specifications depend on the WS-Security specification, making practically necessary to implement it within the DPWS stack of the controller, among other technologies it is dependant like XML Encryption and XML Signature.

Advantages: It is flexible and extensible. It allows end-to-end secure communications. It gives to the designer a deep level of granularity to make fine adjustments according to the application requirements.

Disadvantages: Exchanged messages are verbose due to the fact that the mechanisms used in WS-Security are XML-based; this implies costly processing time for parsing operations. In order to apply XML signature is necessary that the message goes through a canonization process; this is necessary because XML signature validation fails for messages with same content but different structure arrangement.

Applying the security mechanisms established in WS-Security on each monitoring event, can provoke a heavy overload affecting the performance of the device, due to the computational effort of XML Encryption and XML Signature. Fortunately this overload can be reduced by using WS-SecureConversation which is explained later.

WS-Policy

WS-Policy works as a framework that allows to exchange and express the conditions for the interaction between WSs. By using WS-Policy an industrial component can express the security mechanisms it supports, for example, supported cryptographic suites. Then the monitoring server checks those security requirements and follows with the communication process under the security policy alignments.

The reconfigurability of the monitoring system is empowered by using WS-Policy, since new participants offering monitoring services can engage or subscribe to the industrial devices by getting to know which security protocols to follow in an automatic way. This reduces the intervention of human operators in configuration tasks.

***Advantages*:** Implementing WS-Policy within the DPWS stack of an industrial controller not only facilitates the understanding of requirements between the participants for security purposes, but it can be used for negotiations with other objectives like agreement of QoS parameters or automatic selection of monitoring devices based on their policies. Empowers the automatic reconfigurability of the system.

***Disadvantages*:** Interoperability problems can arise if the communicating participants do not share the same semantics for the requirements specified in the policy. This can be solved by using an ontology.

WS-Trust

This specification defines a model that allows to establish a trust relation between two or more entities that do not have prior knowledge of each other. This is done by using a third component which is trusted by the participants. This trusted component is referred as the Secure Token Service (STS). The STS issues security tokens which are used by the participants to protect the information and validate its authenticity.

WS-Trust can be used in the subscription process of external participants (which provide monitoring services) to the industrial devices or in the issuing of monitoring events as depicted in Figure 4.6. In the first step the industrial device request a token to the STS. The STS verifies that the requester is valid and issues a security token. Then the industrial device uses information contained in the token to apply encryption mechanisms in the monitoring events. Finally the now protected message and the token are sent to the monitoring server which corroborates the authenticity of the issuer and the integrity of the data. The gray area in the diagram shows the scope of trust among the entities, usually referred as trust domain.

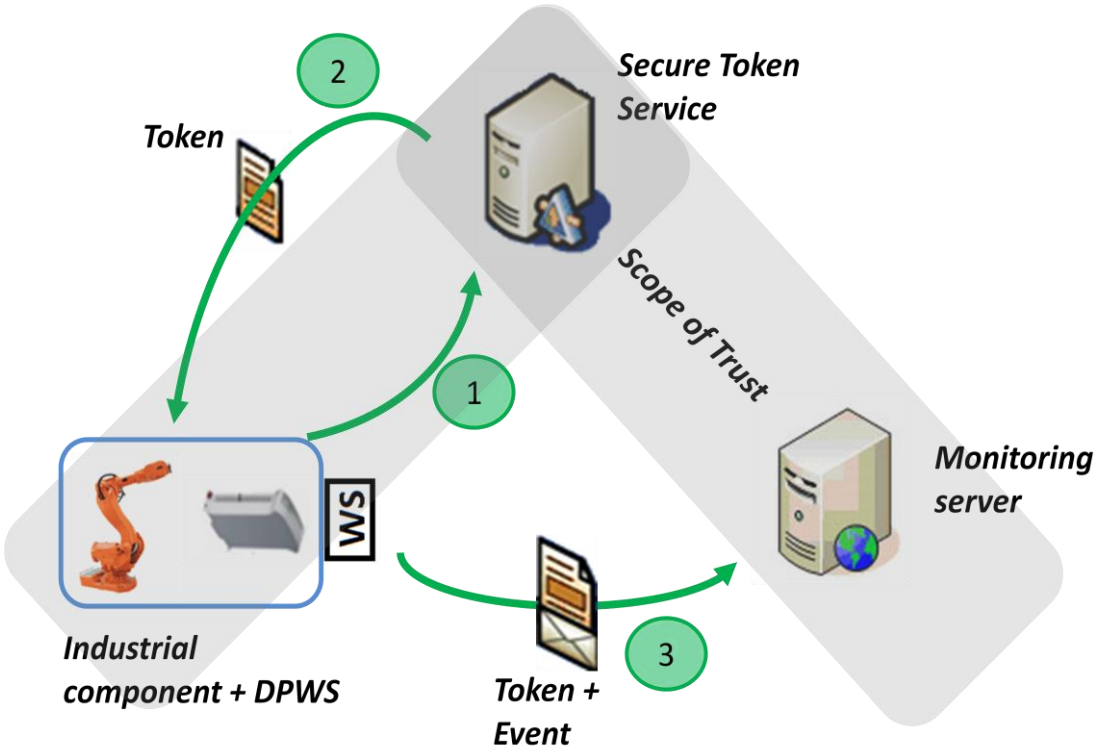


Figure 4.6. *WS-Trust in a monitoring application of industrial components.*

WS-Trust allows to expansion of the scope of trust among different participants located external networks, which is the case for remote monitoring. It is a flexible system and allows different configuration layouts, like indirect trust domain where two STSs establish a trust relation through a third one. It can work with different kind of security tokens, being SAML tokens the most common.

***Advantages*:** It provides a mechanism for federated security, thus allowing the reconfigurability and scalability of the system. Allows the integration of multiple participants. Can work with different type of tokens. It enhances the interoperability of the system.

***Disadvantages*:** Participants must be recognized by the Security Token Service, so prior registration to the service is needed.

WS-Privacy

WS-Privacy provides a framework for expressing information about privacy requirements by combining other specifications like WS-Policy, WS-Security and WS-Trust. It intends to make a straightforward match between the requirements expressed with WS-Policy and the mechanisms provided by WS-Security, using WS-Trust as a mean for evaluating the privacy claims.

The functionality provided by WS-Privacy can be used to speed up the process of deploying a secure monitoring of industrial environments with WSs. It also allows monitoring data providers and consumers to have a common framework to express, evaluate and transport privacy statetments.

Advantage: From the designers point of view, it reduces the labor of matching security requirements (WS-Policy) with the security mechanisms (WS-Security)

Disadvantage: Few formal implementations of this specification. It is tied to other three specifications of the security suite, forcing their implementation in the DPWS stack.

WS-SecureConversation

WS-SecureConversation is an extension of WS-Security. It allows to exchange multiple messages without having to execute authentication process and encryption key agreement for each individual message. This is done by using a security context, in which the same security parameters (encryption keys) are applied to all the messages exchanges within the session. Thanks to this it is possible to reduce the processing load at both endpoints of the communication channel. Once the session expires a new security context has to be established.

By nature, a monitoring application implies a constant flow of messages (monitoring events) from the DPWS unit to the monitoring server. Having to apply individual authentication mechanisms and establish encryption key for each of the messages is computationally very costly, and the performance of the system can be affected severely, especially if it involves resource constrained devices. Fortunately WS-SecureConversation suits for this application and can be used to reduce this data processing overload by creating a secure session for all the monitoring events.

WS-Secure conversation is a flexible specification, which allows different topologies to create the secure context. The simplest case is that one of the participants creates the security context token, then the other participants must decide either to accept it or not. Figure 4.7 shows an example of creating a security context by using a Security Context Token Service. In similar way to WS-Trust, a third component is used issue a security context token, this token is used by the DPWS device to create a security context, the security context token is also transmitted to the monitoring server to know the security parameters of the communication. In this way all the monitoring events, are signed and/or encrypted with the same encryption key (contained in the security context token).

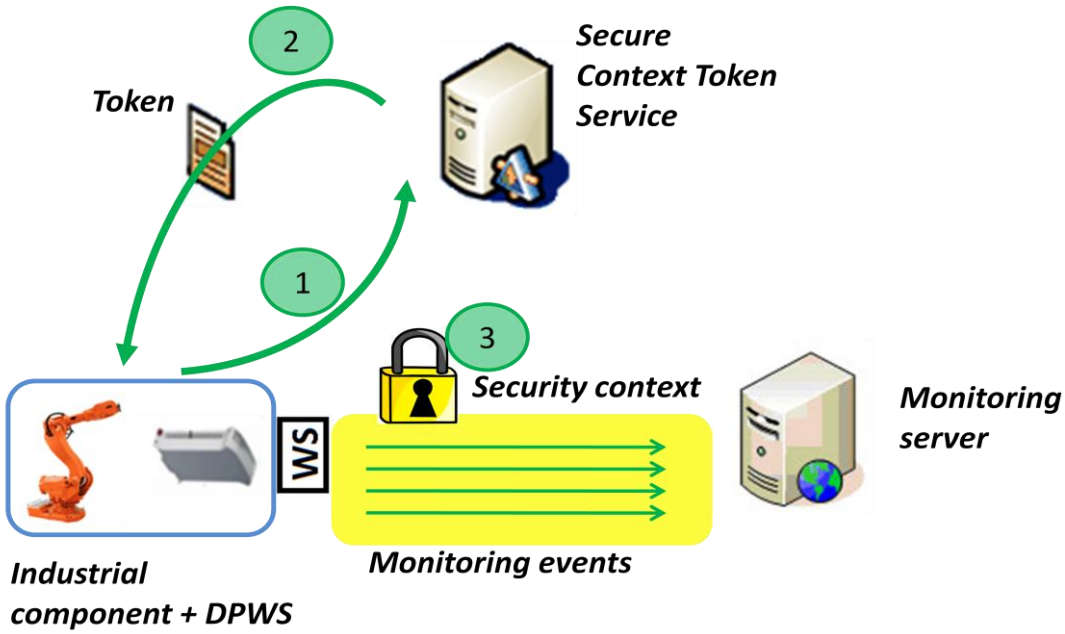


Figure 4.7. *WS-SecureConversation in the monitoring of an industrial component.*

Advantages: Reduces the processing overload in both endpoints of the communication channel. If used with a Security Context Token Service it can be highly scalable.

Disadvantages: The DPWS devices and the monitoring server must trust to the Security Context Service Context prior to the communication exchange.

WS-Federation

WS-Federation is built in top of WS-Trust. It basically has the same objective which is to establish trust domains among entities. The difference is that WS-Federation focuses on trust domains among federated components, like Security Token Service or Identity Providers. Therefore it empowers cross-realms interoperability. WS-Federation makes use of WS-Policy to exchange metadata in order to establish agreements among federated components, each of them supporting different topologies, security and communication requirements.

WS-Federation is thought to empower federated security in very heterogeneous scenarios. For the application of monitoring industrial environments, the scenarios can be considered steady. Therefore the interoperability and security federation requirements can be addressed just by using WS-Trust. In case the heterogeneity of the scenarios increase and become more dynamic WS-Federation could be implemented on top of WS-Trust without disruptive changes. This is one of the advantages of the composeability and interoperability between WSs.

***Advantages*:** Empowers federated security. Allows interoperability among a broad range of trust and communication topologies.

***Disadvantages*:** It depends on WS-Security, WS-Trust and WS-Policy, therefore in order to implement WS-Federation, its dependencies must also be implemented in the DPWS device.

WS-Authorization

As it name implies, WS-Authorization, offers mechanisms to generate authorization decisions in the context of WSs. WS-Authorization defines how the claims contained in the security tokens will be processed and evaluated within the service provider in order to grant or refuse access to some resource. WS-Authorization makes use of authorization policies in order to evaluate those claims.

There has been little development of WS-Authorization compared with XACML. Therefore, for the monitoring application it would be recommended to use a robust and flexible technology like XACML in order to have an authorization mechanism.

***Advantages*:** Addresses authorization mechanisms specifically for WSs.

***Disadvantages*:** WS-Authorization is not widely used. XACML has more popularity, research and development.

4.4 Threats Identification

The application of monitoring an industrial environment with WSs is exposed to different types of threats. The impact that those threats can have on the industrial infrastructure can vary but they can be as severe as putting in risk the integrity of workers and public services. The objective of this section is to identify the different kind of threats that can affect the information assets of the monitoring system or the system itself.

4.4.1 Spoofing Attack

The spoofing attacks are based on creating a fake identity in order to gain access on the network and have privileges over resources. It affects directly the authentication and authorization services. However, if the attack succeeds and the attacker participates in the communications, other threats arise. For example the integrity and confidentiality of the messages that goes through the attacker can be easily compromised or the attacker can do some DoS attack against some specific target.

Monitoring a system with DPWS devices posses an application level vulnerability that can be exploited to execute these attacks. Recapitulating, when a new DPWS device joins a network it goes into a discovery process. In this process a probe match multicast is sent to all the devices within the network in order to discover the available devices, their hosted services and how to access them, all this information contained within the WSDL file. The easiness to perform this discovery and the lack of confidentiality protection uncovered by the DPWS standard (DPWS standard covers integrity and authentication of discovery messages) leaves a door open for the attacker to join the network and get to know all the available devices and their hosted services, after all WSs are meant to be easy accessible.

This can be potentially dangerous, as illustrated in Figure 4.8, once the attacker access the WSDL file, it can get to know all the information required to invoke the services like: IP address where the service is hosted, messages names and structures. All

this data gives the attacker valuable information that can be used to infer communication interactions within the system. In this example the attacker gets to know that there is a control entity, its IP address and its output message, in this case a control message. It also gets to know that there is a valve that can be controlled with a WS invocation and its IP address. Once the attacker gathers this information, it can control the valve pretending to be the control logic orchestrator. In order to increase the chances to succeed, the attacker can change its IP address to the one of the control logic orchestrator, by using IP spoofing, a well known spoofing attack technique. IP spoofing would be useful for the attacker in case the target performs some authorization process based on the IP source. Other types of spoofing are possible for example, if the attacker manages steal data from credentials.

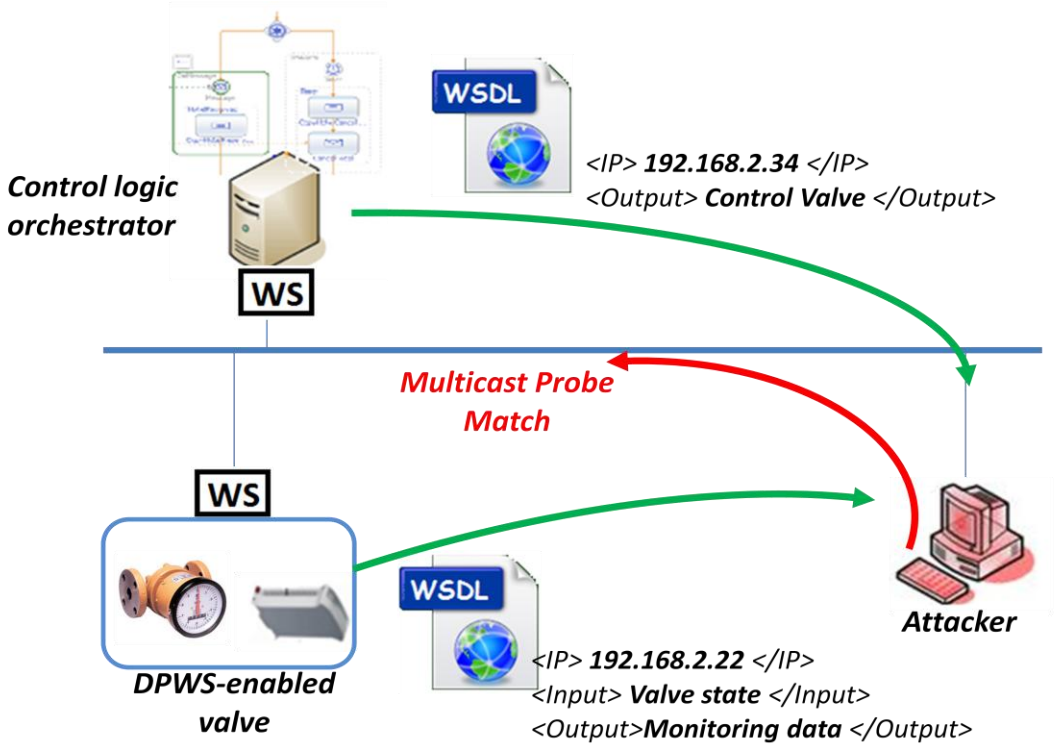


Figure 4.8. *Security vulnerability by using discovery in DPWS devices.*

If the security level to access the industrial network is low, then spoofing attacks can be performed by insiders. Nevertheless if the monitoring application allows remote subscription to the devices, and the filter is based on IP, then an outsider attacker could spoof its IP address to a valid one and gain access on the network. In that case distributed layouts could also be listed as susceptible. So the authentication and authorization mechanisms should not be limited to simple filters like IP filters or username/password, but rather include more secure means like security tokens.

Monitoring applications under the distributed layout can be exposed to a sophisticated spoofing attack. If the system uses 3G network, the attack consists in using a fake base station, in such a way that the 3G modem engages with that fake base station and the traffic is controlled by the attacker. This is a complex attack and the low sensitivity of data in distributed layouts makes it very unlikely to happen.

4.4.2 Eavesdropping Attack

The principle of an eavesdropping attack is to intercept the messages exchanged between two endpoints. Once the message has been intercepted its information can be disclosure or even modified, which would represent a potential hazardous action for the system stability. The most popular schema for eavesdropping is the so called Man in the middle where an attacker makes believe to the endpoints that they are communicating directly between them but actually the communication messages are flowing through the attacker.

If the industrial network uses a hub to connect the devices to each other, the data can be easily disclosure by sniffing in the network. This attack is highly potential in a system that uses DPWS, because of the discovery vulnerability already discussed in the previous point. Eavesdropping threat also arises if the monitoring system includes multiple intermediaries; in that case if sensitive fields of the SOAP message are not properly encrypted, then the data can be disclosed by non intended intermediaries.

Distributed layouts using 3G networks should not consider seriously the eavesdropping of information in the radio link. The information transported by 3G is encrypted with a sophisticated algorithm, although it was recently published a way to disclosure the information, the process requires considerable computational effort and the low sensitivity of the data in distributed layouts makes it unlikely to occur.

4.4.3 Logon Abuse Attack

This threat is possible if the credentials to access a resource are obtained by an attacker, or the authentication and authorization mechanisms bypassed. Once the attacker bypasses them, it has unauthorized access to the resources. This can imply that other threats would be exploited once the attacker manages to access the network or resource.

It is more likely that the organization insiders manage to get to know valuable information that would allow them to do logon abuse attacks. For example operators can get to know the device configuration passwords and make malicious or accidental changes in the device settings.

4.4.4 DoS Attack

It is important to remember that in order for a DPWS device to work as a service provider, it must listen for service requests. This is done by running an embedded web server which receives the WSs requests. Those requests are transported over HTTP. Thus, a DPWS device is exposed to the same DoS attacks as other HTTP servers would be.

The working principle of DoS attacks is very simple, the objective is to make a resource unavailable to its legitimate users by saturating the target with communication requests. Once the target collapses it is unable to provide its service. Networked assets are very vulnerable to this kind of attacks and in order to prevent them, a traffic block-

ing component is required, like firewalls, switches, routers, proxy appliances or server clusters. In cyber security, DoS have shown to be a very simple to execute and effective attack.

The threat of DoS in an industrial environment reaches a more alarming dimension compared with DoS traditional IT infrastructure. When an IT server is under a DoS attack the consequence is the unavailability of information for some period of time. If an industrial device is under a DoS attack, not only the availability of information (for example monitoring data) is affected but also the device loses the capability to attend control requests. For example, Figure 4.9 exemplifies the case of a valve which reports monitoring data through WS events and receives control commands through WS invocations. If an attacker, either insider or outsider, manages to execute a DoS on the valve DPWS interface, it would disable the valve to receive control orders, for example, a "close valve" order would not be processed by the valve which implies potential hazardous effects. This is an example of how DoS attacks in an industrial environment can affect not only information assets but also hardware assets.

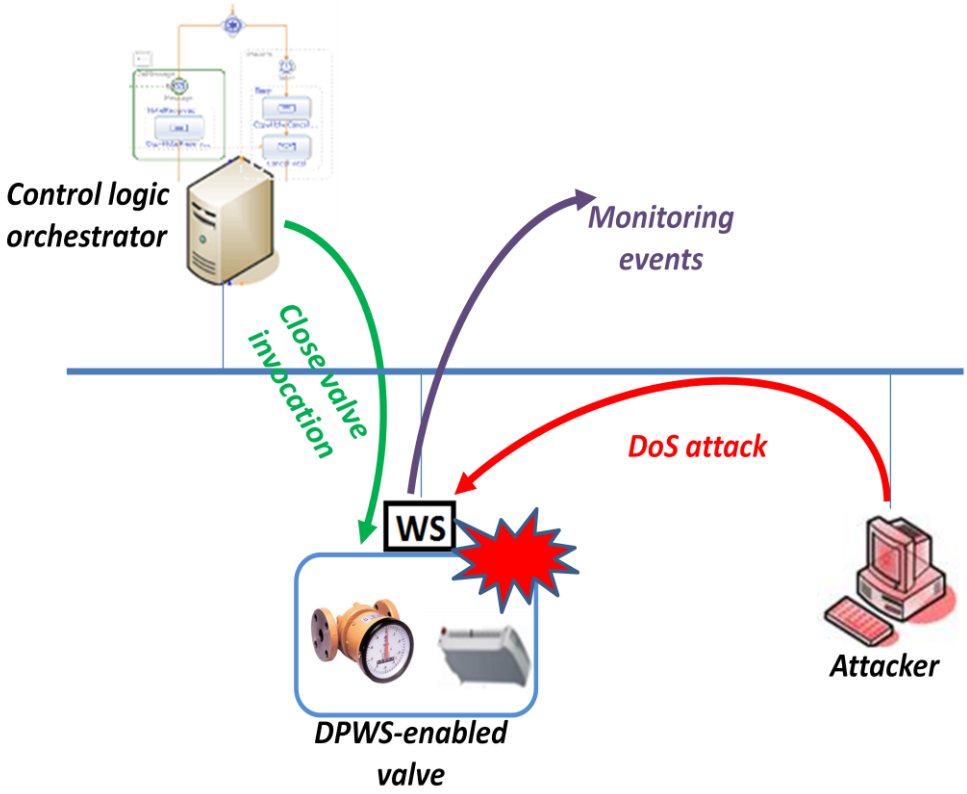


Figure 4.9. *DPWS industrial unit under DoS attack.*

Applications that use DPWS devices have to face the fact that there is available lot of information and ready-to-use applications to execute DoS attacks. Another point to consider is the fact that DPWS devices are resource constrained. Therefore it is easier and requires less effort for a DoS attacker to saturate the computational resources of the embedded server of a DPWS device.

There is a big variety of these attacks, a simple one would be SYN flood, where TCP connections are requested to be open but never acknowledged, which causes the

overflow of the DPWS TCP connections buffer. This is a flaw of the TCP protocol at layer 4 of the OSI model.

The agents of this attack can be insiders and outsiders. It is relative easy for insiders, since they just need a computer connected to the industrial network and this is possible in a monitoring application with a centralized layout. If remote subscriptions to the device are enabled, then outsiders can also execute this attack. But it would require that they bypass the security offered by the DPWS gateway. In this case centralized and distributed layouts are susceptible.

4.4.5 Application Level Attack

Application level attacks are possible due to flaws or vulnerabilities in the application layer of the OSI model. For example, in the case DPWS devices or other component of the monitoring architecture has to validate XML signatures for authentication, there would be threat to suffer Web Service Signature Overflow. In this attack a big amount of nested signatures are placed within the SOAP message. The unit attempting to validate the nested signatures would have to decrypt each by each until it reaches an overflowing point. Again, the low memory of DPWS devices makes them susceptible to this type of attacks.

It is more feasible for insiders to execute this attack but it can be also done by outsiders if the monitoring system allows remote connections for subscriptions.

4.4.6 Radio Jamming

Radio jamming attacks occur by injecting radio signals in a wireless medium. The objective of these signals is to obstruct or block the wireless communication either by inference or by decreasing the signal to noise ratio of the communication channel. This attack falls in the category of DoS attacks but is mentioned separately to remark this kind of attacks on wireless channels.

3G networks can suffer from radio jamming attacks. Actually there are available in the market radio jammers. If a radio jammer is positioned close enough to the industrial equipment, then it wireless interface would be disabled, making impossible for the DPWS-based controller to report the monitoring events. This threat applies for all those wireless communication channels and thus, monitoring applications under a distributed layout are susceptible to it.

Table 4.3 summarizes the threats identified in this section, the security services they affect, the monitoring layout (centralized, distributed) which are susceptible and the agents (insiders, outsiders) that are likely to execute those threats.

Table 4.3. *Threats in the monitoring of industrial environments with WSs.*

Threat	Affected security service	System layout	Agent
Spoofing attack	• Authentication • Authorization • Indirectly others[2]	• Centralized • Distributed[1]	• Insiders • Outsiders[1]
Eavesdropping attack	• Confidentiality • Integrity • Indirectly others[2]	• Centralized	• Insiders
Logon abuse attack	• Authentication • Authorization	• Centralized • Distributed	• Insiders
DoS attack	• Service availability	• Centralized • Distributed[1]	• Insiders • Outsiders[1]
Application level attack	• Service availability • Indirectly others [2]	• Centralized • Distributed[1]	• Insiders • Outsiders[1]
Radio jamming	• Service availability	• Distributed	• Outsiders

[1] In case remote subscriptions to events are allowed
[2] Other security services are not directly affected by the threat but can indirectly be exploited

4.5 Secure Framework for WS-based Monitoring of Industrial Facilities

In order to propose a secure framework for monitoring of industrial facilities with WSs, the previous analyses from this chapter are used. For instance a secure framework must consider the system requirements, its components limitations, the available security mechanisms and the threats to which the system is susceptible.

In cyber security is not possible to provide a unique solution that fits to all the scenarios and that covers all the threats. In order to cyber secure a system it is necessary to focus on its specific requirements. This framework is optimized for the case of monitoring industrial facilities with WSs and it offers a series of guidelines that allow choosing the secure components and mechanisms depending of the specific conditions of the monitoring application. This was done in this way because the monitoring application can have a number of different variants like:

- Monitoring layout: Centralized or distributed
- Reconfigurability or scalability needs
- Required security services
- Policy for remote subscriptions to devices (Allow or deny)
- Monitoring events intended for a single end point or multiple intermediaries

- Amount of monitored DPWS devices (High or low)
- Frequency of generated monitoring events
- Accessibility to the industrial network from others company's networks

A secure framework is composed by architectural components arranged in a specific topology and secure communication protocols used by those components. This section describes the selection of the components and security protocols based on decision diagrams that consider the variables previously mentioned. It is important to mention that the main objective of the secure components architecture is to grant the assets service availability, while the security protocols procure the data integrity, confidentiality, authentication and authorization services.

Both guidelines, for deciding about the architectural components and the security protocols should be applied for every monitoring application. Therefore it is important to identify the application requirements and threats. A good identification of these parameters will result in a better selection of the architectural components and the security protocols.

4.5.1 Architectural Components

The architectural components of the framework must be arranged in a layered topology in order to provide Defense-in-depth security. This is was found as a requirement for protecting industrial infrastructure in previous analyses. As it was already mentioned, the main objective of the architectural components and their arrangement is to protect the assets and grant their availability.

DPWS devices are intended to use a gateway for communicating the monitoring data through internet. It is important that a firewall is used in that gateway and its configuration is properly set.

If the system needs to address reconfigurability and scalability needs, federated security is necessary to provide an easy and flexible management of authentication credentials and authorization policies.

Depending on the monitoring layout (centralized or distributed) different components can be allocated to improve the security. Centralized layouts should include an IDS system within the industrial network with the aim to detect an intrusion or attack within the network. As it was already analyzed in the mechanisms assessment is not practical to place an IDS for every unit deployed in a distributed layout. One important consideration, in the implementation of monitoring applications with distributed layouts, is to restrict the physical access to the industrial infrastructure. This is because the components will be located in remote locations without specialized supervision. If access is restricted properly lot of physical threats can be avoided, like radio jamming.

If the system allows remote subscriptions to the DPWS devices monitoring events, then an access control mechanism is needed. For centralized layouts it is feasible to install a DPWS gateway in the DMZ. The DPWS gateway is just used for subscription

purposes. Monitoring events bypass this element. For distributed layouts is not practical to install a DPWS gateway for each deployed unit. Two alternative possibilities are:

- Forward the requests coming to an specific port to the DPWS device. The device should just allow subscription related operations. The gateway's firewall should use some access mechanism like IP-based, so that just authorized monitoring servers can subscribe to the events.
- Request subscriptions by polling. With this technique, the DPWS device periodically polls from a trusted server, a list of subscribers interested in monitoring events.

As it was studied a big percentage of the threats within an organization are produced directly or directly by the insiders. In order to reduce the risk of compromising the security because of insiders fault, it is highly recommended that the industrial network gets isolated from other networks within the organization by using firewalls and configuring them properly.

Figure 4.10 shows a decision diagram in order to facilitate the selection of the framework architectural components. The blocks in blue color decide the flow of the diagram while the blocks in yellow color set a component or recommendation to include into the framework.

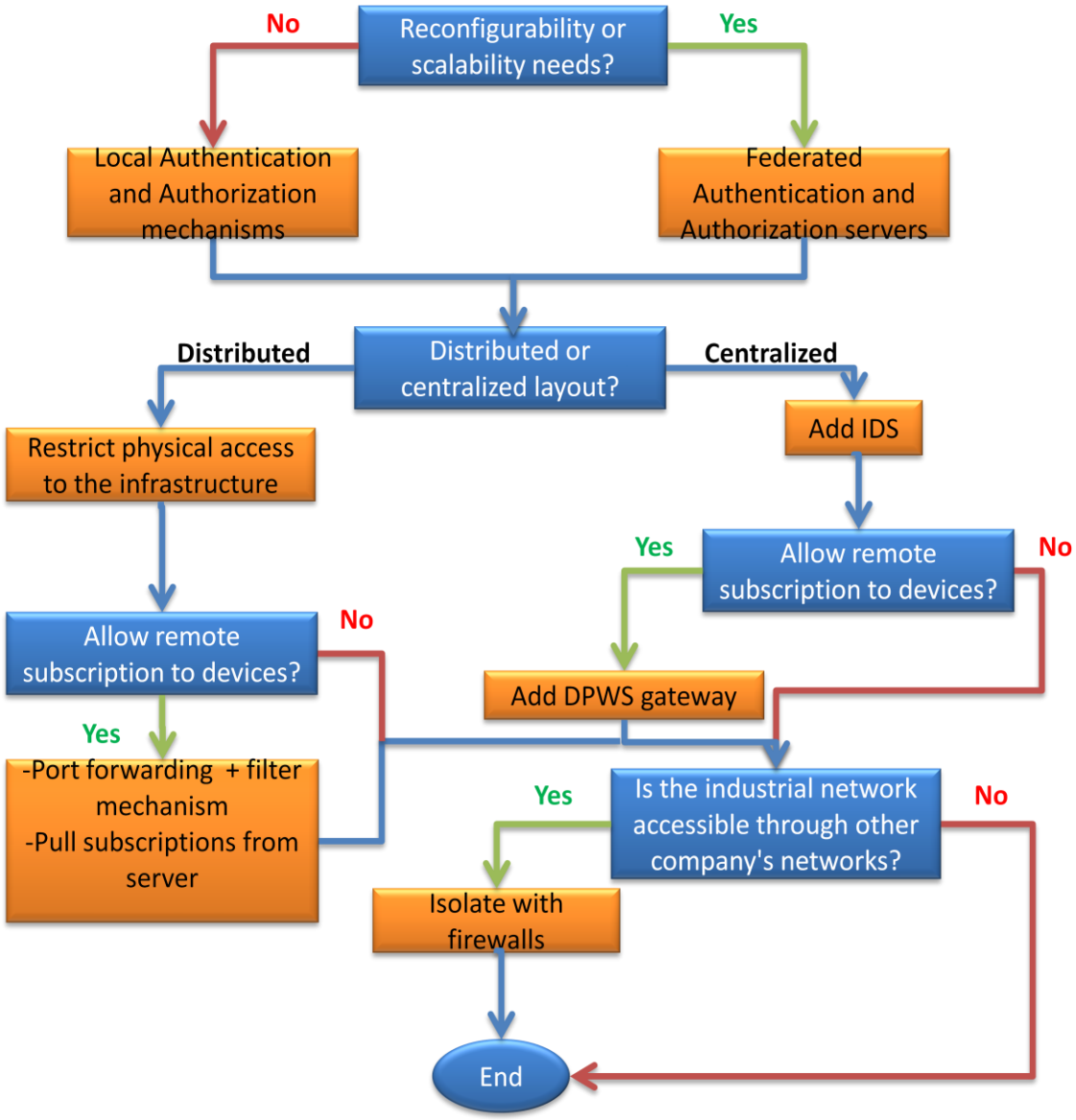


Figure 4.10. *Decission diagram for choosing the architectural components of the secure framework.*

Centralized monitoring example

Consider the case for a centralized monitoring application with the next characteristics:

- Centralized layout for monitoring
- Reconfigurability and scalability needed
- Allowing remote subscriptions to the DPWS devices
- Industrial network accessible from other networks

With these variants, using the decision diagram for the architectural components, the components that should be introduced in the framework, in a security-in-depth arrangement are depicted in Figure 4.11.

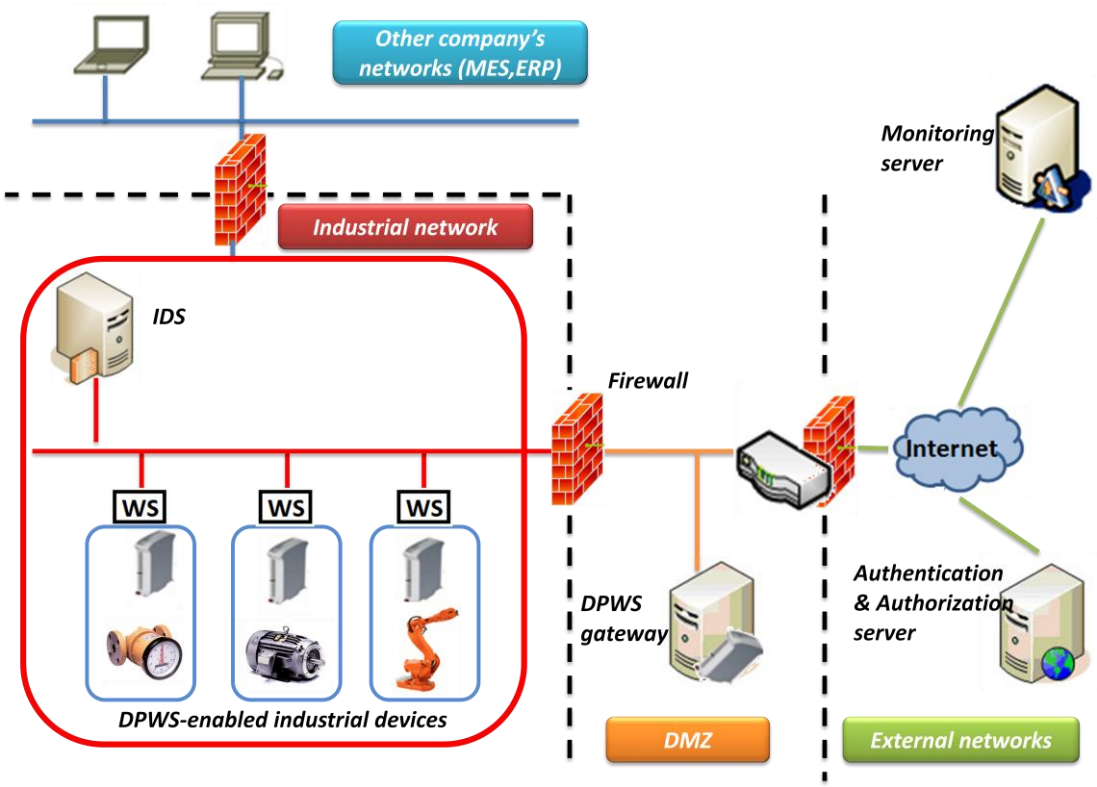


Figure 4.11. *Secure components with Defense-in-depth arrangement for a centralized monitoring case.*

Distributed monitoring example

Consider the case for a distributed monitoring application with the next characteristics:

- Distributed layout for monitoring
- Reconfigurability and scalability needed
- Allowing remote subscriptions to the DPWS devices
- Industrial component isolated from other company's networks

With these variants, using the decision diagram for the architectural components, the components that should be introduced in the framework, in a security-in-depth arrangement are depicted in Figure 4.12.

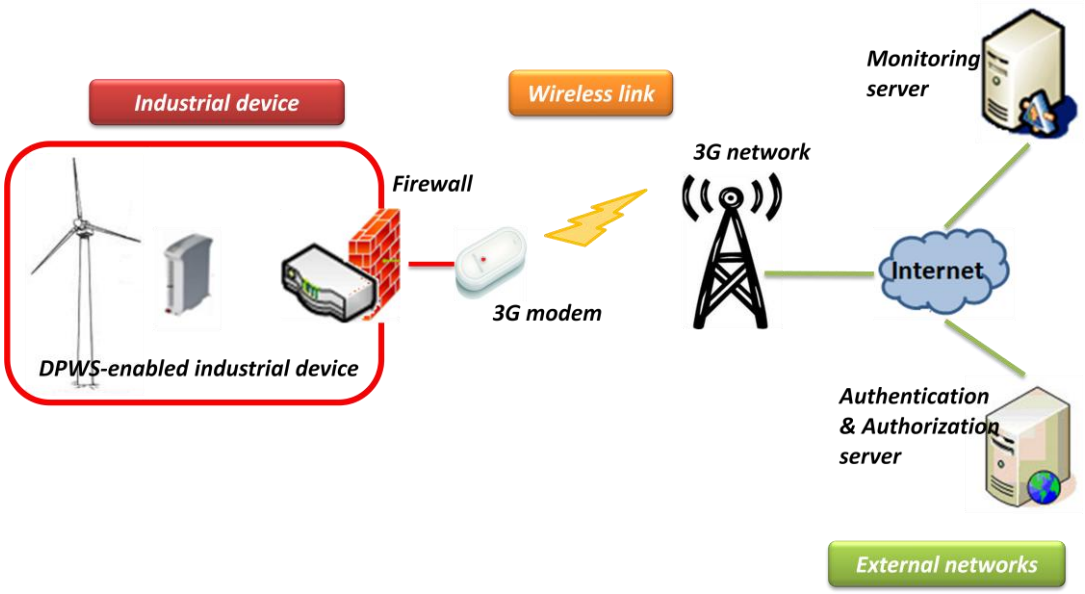


Figure 4.12. *Secure components with Defense-in-depth arrangement for a distributed monitoring case.*

4.5.2 Security Protocols

Security protocols are used to provide data integrity and confidentiality in the exchange of monitoring events. They also offer authentication and authorization services in order to allow or deny access to DPWS devices and their hosted monitoring services. From the security mechanisms assessment section, it is possible to notice that there is a wide variety of security protocols. They should be selected depending on the application requirements and threats to which the system is susceptible.

Once the security protocols have been selected, it is important for the implementer to ensure interoperability among them. This can be a serious technical problem, if the security protocols are not flexible or not meant to be interoperable with other protocols at different OSI layers. A big advantage of using components from the WS Security suite, is that they have been designed with composeability in mind, thus interoperability is easily achieved between them.

The monitoring application can have two types of end points: single end points and multiple intermediaries. In single end points, all the events generated by the DPWS devices are gathered by one monitoring server. In multiple intermediaries, the events generated by the DPWS devices are supposed to be processed by multiple intermediaries. This implies the forwarding of the monitoring event through different intermediaries.

If the monitoring events are intended for a single end point and reconfigurability and/or scalability are not needed, then data integrity and confidentiality by securing the message at transport layer. TSL/SSL is chosen as the transport layer security mechanisms; the reasons are detailed in the security mechanisms assessment section.

Considering still the case where the monitoring events are intended for a single end point; if authentication and authorization services are required; two schemas can be adopted: local or federated. Local mechanisms should be adopted if the amount of

DPWS devices that are monitored is low and not scalability is required. Local authentication and authorization mechanisms use lists contained within the DPWS devices/monitoring server with participants with are authorized to use their services. Local authentication/authorization mechanisms can include the use of credentials like username/password and/or the use of shared private keys for encryption purposes. A federated approach should be used if the amount of DPWS devices that are monitored is high or scalability is required. Then the implementation of servers and DPWS devices with SAML and XACML capabilities is needed.

If the monitoring events are intended for a single end point but the reconfigurability and scalability of the system are required; or the monitoring events are intended for multiple intermediaries, then security at message level is highly suggested. Leveraging the benefits of using WSs as the core communication technology of this application, some of the specifications of the WS security suite are recommended.

In order to reduce interoperability conflicts and facilitate an automatic agreement between the communicating parties, WS-Policy should be implemented. WS-Policy allows the interacting parties to communicate what kind of security requirements they have and which cryptographic suites they support.

The dynamic of the monitored system has a direct effect in the amount of monitoring events generated by the DPWS devices. If there is a frequent generation of events, WS-SecureConversation should be used to avoid the overload caused by executing security processes for each individual message. WS-SecureConversation intends to boost the system performance, so it should also be tested/used if the application presents slow performance when using cryptographic operation for providing security services.

If authentication and authorization services are required; two schemas can be adopted: local or federated. Local mechanisms should be adopted if the amount of DPWS devices that are monitored is low and not scalability is required. A federated approach should be used if the amount of DPWS devices that are monitored is high or scalability is required. Then the implementation of servers and DPWS devices with SAML and XACML capabilities is needed. Leveraging the use WSs, WS-Trust is used as a transport mechanism in order to transmit the SAML tokens.

If WS-SecureConversation or WS-Trust are used, then it is necessary to implement WS-Security because they are built on top of it. WS-Security should also be implemented if data integrity and/or confidentiality services are needed.

Figure 4.13 resumes the guidelines presented in this section in a decision diagram. This decision diagram allows choosing the security protocols depending on the monitoring application characteristics. The blocks in blue color decide the flow of the diagram while the blocks in yellow color define the security protocol that must be implemented and included by the framework.

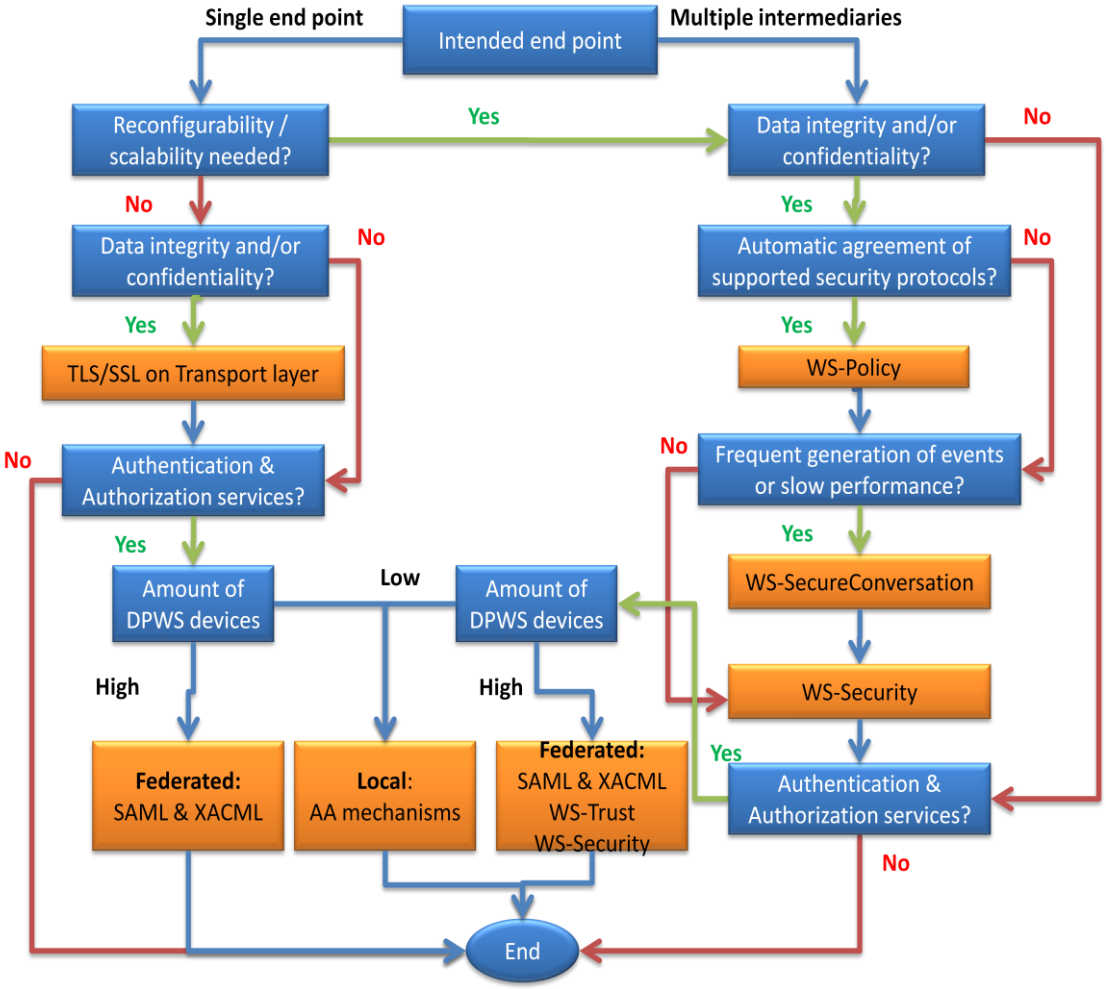


Figure 4.13. *Decission diagram for choosing the security protocols of the secure framework.*

It is important to highlight the fact that the decision diagram for security protocols does not consider the type of monitoring layout (centralized or distributed). This is because the communicating parties in both monitoring layouts are the same. Both of them involve DPWS devices and monitoring servers. So indistinctly of the monitoring layouts, the communicating parties should implement the corresponding security protocols.

Monitoring application example

Consider a monitoring application with the next requirements:

- Monitoring events intended for a single end point
- Reconfigurability or scalability needs
- Integrity, confidentiality, authentication and authorization services required
- Automatic agreement of security protocols between communicating parties
- High amount of monitored DPWS devices
- Frequent generation of monitoring events

With these variants, using the decision diagram for the security protocols, the security protocols that must be supported by the framework are:

- WS-Policy
- WS-SecureConversation
- SAML
- XACML
- WS-Trust
- WS-Security

These protocols ensure that security requirements are addressed. It is necessary to create a security profile that includes them and that is compliant with the DPWS. The recommended protocols are message level protocols and highly interoperable with WSs.

5 FUTURE WORK

The objective of this chapter is to mention the gaps and opportunity areas that were identified during the development of this work. Although there is enough research and implementation of security for WSs, there is very little done for WS security at device level, like DPWS devices. If it is truth that lot of the techniques and frameworks that are applied to WSs can be applied to WSs when they are hosted in devices, it is also a fact that in some cases they cannot be applied seamlessly to industrial communication systems because they have different requirements and conditions than traditional IT systems.

5.1 Implementation of Security in DPWS Devices

It is a fact that the research efforts have concentrated in making functional the deployment of applications that use DPWS devices. Other projects have been developed in order to highlight the advantages of using WSs at device level by applying existing technologies, like WSs orchestration or composition. Functionality and advantages has been demonstrated by proof of concept in different test beds, but in order to be closer to have DPWS devices in a commercial stage, ready to use in domotics or industrial systems, is necessary to provide security services on them.

During the time of this writing, no physical implementation of security protocols has been done in DPWS devices. As it was already mentioned, the efforts are concentrated in other branches. The resource constrained nature of DPWS devices makes it harder for engineers and researchers to implement security protocols available as commercial off-the-shelf (COTS) components.

Another reason that can slow down the integration of security protocols in DPWS devices is the few choices that researches have when choosing an open source platform to implement the DPWS stack in embedded targets.

5.2 Security Code Generator for DPWS Targets

Implementing security protocols can be a tedious task. The implementer has to ensure the interoperability among the protocols dependencies and the code compliance with the embedded target. One way to reduce the complexity of this task would be by using a code generator as shown in Figure 5.1.

The code generator would receive as inputs:

- ***Security protocols to include***: the security protocols and their options can be chosen by the user. If the generator allows flexibility and low granularity level

when selecting the protocols and its options, then is easier for the user to chose specific properties for his/her application. For example, a user might choose to include WS-Security→Data integrity→using Elliptic Curve Cryptography. This low level granularity would result in better fit and less amount of code.

- ***Deployment target***: by choosing the computational target where the DPWS stack is implemented, it would be possible to generate the source / machine code, so that it can be downloaded as firmware for a specific device.

Having a security code generator would reduce significantly the amount of time needed to deploy a DPWS application with security services. The facility to change the security protocols and its variants would make possible to test different configurations to see how they affect the system performance; with the aim to find an optimal compromise.

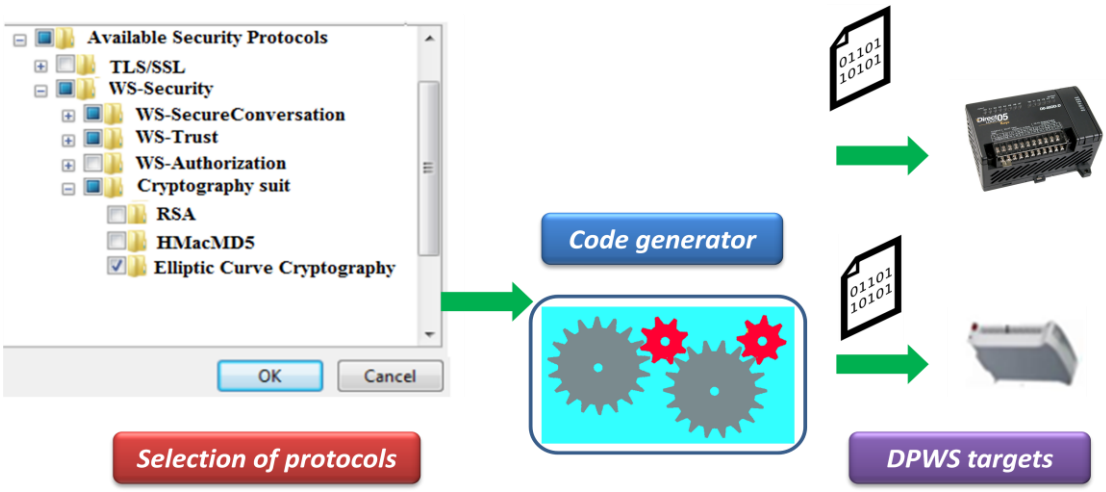


Figure 5.1. *Security code generator for DPWS targets.*

5.3 Performance Analysis in an Industrial System

An industrial system that employs networked embedded controllers and at the same time provides security services will have a performance impact in its components. This is because security services rely in encryption algorithms which are computationally heavy and the resource constrained nature of embedded devices.

It is important to realize tests or models in order to understand how implementing security services impact the performance of the industrial system. The impact can be analyzed at device level and at system level. It would be very useful from the designer's point of view to know how different configurations of security requirements affect the system performance.

In order to create a performance analysis it is necessary to identify Key Performance Indicators (KPI) for an industrial system, its components and its communication system. It is necessary to recognize what KPIs are representative. For example, the overall industrial system can be evaluated with the Overall Equipment Effectiveness (OEE), an industrial controller performance can be measured with processing times for pars-

ing/encryption tasks and the communication system can be evaluated with communication delays. Once the KPIs have been identified then the technical challenge of measure them arises. Later the results should be gathered and interpreted.

Having a proper performance analysis can be used for further studies. For example, it can be used to identify bottleneck points in the system in order to focus efficiency improvements on them. The behavior of different configurations can be analyzed in order to find an optimal compromise for the system. Or it can be used for feasibility studies about implementing security in embedded devices used in industrial systems.

5.4 Simulation of Secure Transactions in an Industrial System

The implementation of security protocols in embedded devices for further performance tests and analyzes can be a very costly, tedious and time consuming task. It involves the programming of the protocols, ensuring interoperability among them, setting the test bed and running the test. In order to avoid all this complexity and speed up the performance analyzes, simulation tools must be used.

A simulation tool would allow the designer to choose the components (DPWS devices, servers, firewalls) of an industrial system, their arrangement and transactions. Then he/she would be able to set security characteristics among those transactions. Then by running the simulation it would be possible to estimate the processing time that takes for each component to process a message with security characteristics. The industrial network and the overall industrial system performance could be forecasted as well. Different configurations/arrangements of components and security protocols can be tested in order to find a proper compromise.

The success of the simulation tool does not rely in implementing the security protocol itself, but rather in estimating the processing time it takes for the device to execute the algorithms derived by the protocol. A good model that estimates the processing times is the technical challenge and of course the model changes depending on the device target.

Simulation is widely used in the IT world for forecasting the behavior and performance of a communication network, and it can be used with the same purposes for an industrial system that uses technologies derived from the IT field. There is lot of background and information available, the challenge consist in including the models that characterize the industrial system.

The control logic of an industrial system that uses DPWS devices can be modeled by using Business Process Execution Language (BPEL). This can be very useful for a simulation tool, because a BPEL file defines the relations and interactions among the WSs available in the system. This BPEL description can be used by the simulation tool as a model of the industrial communication system. Then the designer can specify security services among the interactions. The simulation can be run with different settings for a later comparison of the performance results.

Finally, it is important to remark that is inconceivable to test security protocols on an industrial system which is in operation stage. That involves high risks that can provoke hazardous results in the system and personnel integrity. It is also unfeasible to create a test bed that replicates the components and architecture of the industrial system that is why simulation tools are needed.

5.5 Attack Signatures for Industrial Systems

The proper functioning of an Intrusion Detection System (IDS) relies on the database of attack signatures it uses to analyze/compare with the network's traffic and behavior. If the communication or system behavior pattern matches one of the attack signatures then a security alarm is triggered. There are a wide number of security companies generating and identifying attack signatures for traditional IT systems. Attack signatures for industrial systems have not been developed at all. The lack of information and comprehension of how these attacks profile in an industrial environment might be a reason for this.

One of the risks of using a wrong or not well tuned attack signature, is that the amount of false positive and false negative alarms can be too high. Meaning that the personnel responsible of the industrial system will receive false alarms of attacks or no alarms when the system is under real attacks. In the former case this would imply unnecessary revision and probably stop of the industrial facility, while in the later case the threat can risk the integrity of the personnel and the system itself.

In order to define more precisely an attack signature for an industrial system that uses WSs at device level, the BPEL file of the system can be used. The BPEL describes the relations and interactions among the different WSs which are available in the system. At some extent, the BPEL model can be used to forecast the expected traffic, messages and communication channels between the system components. If the traffic/communication within the industrial network diverges from what is expected according to the BPEL model then an attack could be under execution. The technical challenge relies in how to estimate the divergence between the expected behavior and the real behavior of the industrial network and its components.

6 CONCLUSIONS

Industrial systems encompass critical infrastructure like production systems or public services. It is a first priority to secure the all the assets of the industrial systems (physical and logical), since the compromise of any of them can risk the integrity of the personnel and the system itself.

On the other hand, WSs are an emerging technology in the field of automation systems. Interoperability, reconfigurability, scalability and security can be inherited when WSs are used as the core communication technology to interconnect industrial devices and applications. Nevertheless lots of the research efforts have been concentrated on boosting and optimizing properties like interoperability, reconfigurability and scalability leaving aside security.

This situation might be due to the fact that this technology is going through a Hype Cycle behavior. When a new technology, with high expectations, is introduced it usually evolves in a Hype Cycle manner; this might be the case of WSs in industrial systems. The starting of the Hype Cycle is full of expectations and lots of possible scenarios are visualized. So considering this model, security in WSs for industrial systems will be tangible until the technology goes in the *slope of enlightenment* stage. This stage occurs after the real possibilities of the technology are explored and it becomes more stable and mature.

In security terms, it was found that event-based monitoring offers great characteristics, when compared with traditional polling-based monitoring. It can be easily adapted to defense-in-depth schemas (widely used in industrial cyber security), since outgoing traffic can pass without difficulties through firewalls. By using WSs events, it is possible to leverage the security mechanisms and frameworks used in WSs. One of these advantages is the implementation of security at message level, which boosts the interoperability and reconfigurability of the monitoring system.

In order to provide cyber security on an industrial system, it is important to know its conditions and infrastructure. Most likely, an industrial environment is deployed under a centralized or distributed layout. The conditions and infrastructure of centralized deployments offer good characteristics for the enhancement and maintenance of cyber security. Nevertheless the data sensitivity can be considered high and therefore more susceptible to suffer attacks. Bottlenecks can be another complexity of this type of deployments due to possible high density of devices interfaced to an external network through a single gateway. On the other hand, the nature of distributed layouts makes unpractical to implement extra security appliances on each node. However the data coming from single nodes is less sensitive.

When designing an industrial system, it is important to consider its scalability and reconfigurability requirements. These requirements will have an impact in the cyber security architecture of the system. For instance, if monitoring service providers need to engage dynamically to the devices located in a remote industrial facility. Then, it is necessary to offer a mechanism that allows external subscriptions to the devices events; a DPWS proxy is proposed for this purpose.

Security requirements of industrial systems differ from those of traditional IT applications. For instance, for the specific case of industrial monitoring, it was found that data integrity and authentication services have the highest priority. Therefore, the selection of security mechanisms should be done with the aim of warranting these two services. Data confidentiality and authorization are second priority services; but of course, it depends on the application. If the data coming from the industrial environment has high sensitivity, then data confidentiality is necessary. If monitoring service providers will subscribe to the industrial devices events, then authorization services are required.

There is a huge amount of security algorithms, frameworks and mechanisms that are used in IT cyber security. Unfortunately security-related processes are computationally demanding and therefore it is not possible to use them seamlessly at device level due to the fact that DPWS devices/controllers are resource constrained. This fact forces designers to select and tune the security mechanisms according to the application needs. Fortunately, the DPWS stack is enough flexible in security terms, since it allows the use of different technologies and it is highly interoperable with security mechanisms offered by the WS Security suite.

D*efense-in-depth* security relies heavily in the construction of shells around the protected assets. Security appliances like firewalls offer a very good and low cost mechanism to achieve this. IDS can be used to increase the strength of these shells and should be deployed if the system conditions allow it.

Among all the security mechanisms candidates that can be used in DPWS devices, those who empower a federated security schema are desired. Federated security allows to isolate the authentication and authorization processes from the DPWS devices, by using a trusted server to perform these tasks. This is wished since these tasks can be computationally demanding for resource constrained devices.

It is also desired to preserve message level security in order to have the benefits of end-to-end security, like scalability and reconfigurability. For this reason, tunneling protocols or mechanisms that secure data at the transport layer are avoided.

Without doubt, one of the most outstanding properties of the DPWS stack is its alignment and interoperability with WS specifications. This opens the possibility to make use of the WS Security suite in DPWS devices. The WS Security suite has two main characteristics. First, all the security mechanisms are applied at message level, thus is possible to inheritance advantages like: end-to-end security, selective trust of domain, transparency, easiness for message transference and communications flexibility and interoperability. Second, it is a fine grained pool of security-related protocols. This

allows the designer to choose and configure specific security mechanisms in order to address the application needs.

An application that uses the WS Security suite has to implement as fundamental specification the WS-Security. Above this, other secure specifications can be used, for instance, the nature of monitoring applications makes WS-SecureConversation suitable for reducing processing overload or WS-Trust can be used to implement of a federated security schema.

The use of DPWS devices in an industrial environment not only exposes the system to many of the cyber threats and attacks known in the IT; in fact, the resource constrained nature of the devices makes them more fragile against common attacks like DoS because the network buffer of a DPWS device can be saturated much faster when compared with a conventional server. During this work it was encountered that the discovery process of DPWS devices makes the system components and structure transparent to malicious insiders, facilitating further cyber attacks. The isolation of the industrial network with blocking traffic elements, like firewalls, is a must.

In cyber security it is not possible to have a framework (security components and protocols) that fits and covers all the systems against an undetermined number of threats. It is necessary to arrange and tune the security framework depending on the characteristics and needs of the industrial system that needs to be protected. This thesis proposes a set of guidelines, in the form of decision diagrams, that will help the designer to select the architectural components and security protocols for a given industrial application.

The implementation of security protocols in an industrial system has an impact in its performance. It is necessary to evaluate different configurations and tuning parameters in order to find an optimal framework. The real implementation of security components and protocols in an industrial facility for comparison purposes is not practical. Therefore it is necessary to make use of models and simulations in order to choose the one that has the smallest performance impact and still accomplishes the security requirements.

Adopting open communication technologies, like WSs, in industrial systems brings a broad set of benefits; however, it also opens the door to many of the threats and cyber attacks known in the IT world. This is a new situation that the industrial community has to face, since for years, the systems were protected by *security through obscurity*. By using proprietary protocols, little known and not available to community and by isolating the industrial system from external networks was considered enough secure. However nowadays, industrial systems employ open communication protocols and are using public networks, like Internet, to share their data; all of this forcing the evolution of industrial cyber security.

7 REFERENCES

Alonso, Gustavo. *Web Services: Concepts, Architectures and Applications.* Berlin: Springer, 2004.

Armstrong, Randy, and Paul Hunkar. "The OPC UA Security Model For Administrators." OPC Foundation, 2010.

Benatallah, Boualem, Fabio Casati, Daniela Grigori, H. R. Motahari Nezhad, and Farouk Toumani. "Developing Adapters for Web Services Integration." *In Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE 05).* Porto, 2005. 415-429.

Bohn, Hendrik, Andreas Bobek, and Golatowski Frank. "SIRENA-Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains." *In International Conference on Networking (ICN'06).* Mauritius, 2006.

Candido, Goncalo, Francois James, Jose Barata de Oliveiera, and Armando W.Colombo. "SOA at Device level in the Industrial domain: Assessment of OPC UA and DPWS specifications." *In Proceedings of the 8th IEEE International Conference on Industrial Informatics (INDIN-2010).* OSAKA: IEEE, 2010. 598-603.

Cannata, A., M. Gerosa, and M. Taisch. "SOCRADES: a Framework for Developing Intelligent Systems in Manufacturing." *Industrial Engineering and Engineering Management (IEEM 2008). IEEE International Conference on.* Singapore: IEEE, 2008. 1904-1908.

Chen, Jyh-Cheng, and Tao Zhang. *IP-Based Next-Generation Wireless Networks.* New Jersey: John Wiley & Sons Inc., 2004.

Chollet, Stephanie, Philippe Lalanda, and Andre Bottaro. "Transparently adding security properties to service orchestration." *Advanced Information Networking and Applications - Workshops (AINAW 2008). 22nd International Conference on.* OKINAWA: IEEE Computer Society, 2008. 1363-1368.

Cui, H, R Lara, R Makar, N Moelholm, and F Rodrigues. *IBM WebSphere Application Server V7.0 Web Services Guide.* New York: IBM Redbooks, 2009.

de Deugd, S., R. Carroll, K.E. Kelly, B. Millett, and J. Ricker. "SODA: Service-Oriented Device Architecture." *Pervasive Computing, IEEE Volume 5.* IEEE Computer Society, 2006. 94-96.

Delamer, Ivan M., and Jose L. Martinez Lastra. "Information Security for Reconfigurable Manufacturing Systems using Networked Embedded Controllers." *lnformation Control Problems in Manufacturing - INCOM 2006.* St Etienne: ELSEVIER, 2006. 129-134.

—. "Loosely-coupled Automation Systems using Device-level SOA." *Industrial Informatics, 2007 5th IEEE International Conference on* . Vienna: IEEE, 2007. 743-748.

Dzung, D., M. Naedele, T.P. Von Hoff, and M. Crevatin. "Security for Industrial Communication Systems." *PROCEEDINGS OF THE IEEE, VOL. 93, NO. 6.* Baden: IEEE, 2005. 1152-1177.

Erl, Thomas. *Service-Oriented Architecture: Concepts, Technology, and Design.* Crawfordsville: Prentice Hall, 2009.

Falliere, Nicolas, Liam O Murchu, and Eric Chien. *W32. Stuxnet Dossier.* California: Symantec Corporation, 2010.

Gupta, Vipul, et al. "Sizzle: A standards-base end-to-end Security Architecture for Embedded Internet." *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on* . California: IEEE Computer Society, 2005. 247-257.

Harada, Mitsuo. "Security Management of Factory Automation." *SICE, 2007 Annual Conference* . Takamatsu: IEEE, 2007. 2914-2917.

Hatala, Marek, Ty Mey, and Ashok Shah. "Federated Security: Lightweight Security Infrastructure for Object Repositories and Web Services." *Next Generation Web Services Practices, 2005. NWeSP 2005. International Conference on.* Seoul: IEEE Computer Society, 2005. 6-11.

Hernandez, Victor, Lourdes Lopez, Oscar Prieto, Jose-F Martinez, Ana-B Garcia, and Antonio Da Silva. "Security Framework for DPWS Compliant Devices." *Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09. Third International Conference on* . Athens: IEEE Computer Society, 2009. 87-92.

Hosoya, Kensuke, and Hiroshi Miyata. "Applying Security Architecture to Industrial Network Protocol." *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on* . Osaka: IEEE, 2010. 443-448.

IEEE PRESS. "Network Security Current Status and Future Directions." By Christos Douligeris and Dimitrios Serpanos. New Jersey: John Wiley & Sons, Inc., 2007.

Inico. "S1000 User Manual." 2010.

InnoQ. "Web Services Standards Overview." *InnoQ Resources.* 2007. http://www.innoq.com/resources/ws-standards-poster (accessed November 23, 2010).

ISA. "Cyber Security Risk Assestment for Automation Systems." *Industrial Network Security Web Seminar Series.* ISA, 2008.

—. "ISA 99, Industrial Automation and Control System Security." *The International Society of Automation.* 2010. http://www.isa.org/MSTemplate.cfm?MicrositeID=988&CommitteeID=6821 (accessed July 28, 2010).

—. "Security for Industrial Automation and Control Systems Part 1: Terminology, Concepts, and Models." *ANSI/ISA-99.00.01*. North Carolina: ISA, 2007.

Jammes, Francois, Antoine Mensch, and Harm Smit. "Service-Oriented Device Communications Using the Devices Profile for Web Services." *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on* . Niagara Falls: IEEE, 2007. 947-955.

Jang, K., et al. "3G and 3.5G Wireless Network Performance Measured from Moving Cars and High-Speed Trains." *Proceedings of the 1st ACM workshop on Mobile internet through cellular networks (MICNET 09).* Beijing: MICNET, 2009. 19-24.

Karnouskos, S., T. Bangemann, and C. Diedrich. "Integration of Legacy Devices in the Future SOA-based Factory." *In INCOM 09 the 13th IFAC Symposium on Information Control Problems in Manufacturing.* Moscow: ELSEVIER, 2009. 2109-2114.

Komoda, Norihisa. "Service Oriented Architecture (SOA) in Industrial Systems." *Industrial Informatics, 2006 IEEE International Conference on* . Singapore: IEEE, 2006. 1-5.

Krutz, Ronald. *Securing SCADA Systems.* Indianapolis: John Wiley & Sons Inc., 2006.

Lastra, J.L.M., and M. Delamer. "Semantic web services in factory automation: fundamental insights and research roadmap." *Industrial Informatics, IEEE Transactions on.* Tampere: IEEE Computer Society, 2006. 1-11.

Mahboob, Athar, and Junaid Zubairi. "Intrusion Avoidance for SCADA Security in Industrial Plants." *Collaborative Technologies and Systems (CTS), 2010 International Symposium on* . Chicago: IEEE, 2010. 447-452.

Martinez, Jose-F., et al. "A security architectural approach for DPWS-based devices." *CollECTeR Iberoamerica 2008 conference.* Madrid: CollECTeR Iberoamerica, 2008.

Mathes, Markus, Steffen Heinzl, and Bernd Freisleben. "Towards a Time-Constrained Web Service Infrastructure for Industrial Automation." *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on* . Hamburg: IEEE, 2008. 846-853.

Mogollon, Manuel. *Cryptography and Security Services: Mechanisms and Applications.* London: IGI Global, 2007.

Moreira Sa de Souza, Luciana, Patrik Spiess, Dominique Guinard, Moritz Köhler, Stamatis Karnouskos, and Domnic Savio. "SOCRADES: A Web Service Based Shop Floor Integration Infrastructure." *Lecture Notes in Computer Science, 2008, Volume 4952, The Internet of Things.* Berlin: SPRINGER, 2008. 50-67.

Muller, Andreas, Holger Kinkelin, S Kumar Ghai, and Georg Carle. "A Secure Service Infrastructure for Interconnecting Future Home Networks based on DPWS and XACML." *Proceedings of the 2010 ACM SIGCOMM workshop on Home network.* New Delhi: ACM, 2010. 31-36.

Naedele, Martin. "Standarizing Industrial IT Security - A First Look at the IEC approach." *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on.* Catania: IEEE, 2005. 863-869.

Newcomer, Eric, and Greg Lomow. *Understanding SOA with Web Services.* Munich: Addison-Wesley, 2007.

OASIS. "Devices Profile for Web Services Version 1.1." *OASIS: Advancing Open Standards for the Global Information Society.* July 2009. http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.html (accessed July 26, 2010).

Oda, Shinji. "Latest Trend of Industrial Network Security Standards and Vendor Activities." *SICE-ICASE, 2006. International Joint Conference* . Busan: IEEE, 2006. 161-164.

Polk, Wade, Paul Malkewicz, and Jaroslav Novak. "Industrial Cyber Security: From the Perspective of the Power Sector." *DEFCON 2010.* Las Vegas: DEFCON, 2010.

Prakash, Sutirtha, and Sachikanta Behera. "Study and Implementation of 3G Mobile Security." *Bachelor's thesis, National Institute of Technology.* Roukela, 2010.

SAMIA. *Service-bAsed Monitoring for Industrial Ambients: Project.* 2009. http://www.samia-project.info (accessed October 25, 2010).

Shah, Shreeraj. *Hacking Web Services.* Boston: CHARLES RIVER MEDIA, 2007.

Slay, J., and M. Miller. "Lessons learned from the Maroochy water Breach." In *Critical Infrastructure Protection*, by Eric Goetz and Sujeet Shenoi, 73-82. New York: Springer, 2007.

SOASPECS. "WS-* Specs." *SOA Specifications.* 2009. http://www.soaspecs.com/ws.php (accessed October 20, 2010).

SOCRADES. *Service-Oriented Cross-layer infRAastructure for Distributed.* 2009. http://www.socrades.eu/ (accessed July 28, 2010).

SODA. *Security Requirements.* Project document, The SODA Consortium, 2007.

Stephanou, T. "Assessing and exploiting the internal security of an organization." SANS Institute, 2001.

Suomalainen, Jani. "Towards Fine-Grained Authorizations in Small Office and Home Networks." *Systems and Networks Communications, 2007. ICSNC 2007. Second International Conference on* . Cap Esterel: IEEE Computer Society, 2007. 66-71.

Uskela, Sami. "Key Concepts for Evolution Toward Beyond 3G Networks." *Wireless Communications, IEEE Volume 10.* IEEE Communications Society, 2003. 43-48.

Villasenor Herrera, Vladimir, Angelica Nieto Lee, and Jose L. Martinez Lastra. "A methodology for Identifying Web Services in Electronic Manufacturing Systems." *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on* . Osaka: IEEE, 2010. 571-577.

W3C. "SOAP Version 1.2." *World Wide Web Consortium (W3C).* 2007. http://www.w3.org/TR/soap12-part0/#INTRO (accessed October 16, 2010).

Weerawarana, Sanjiva, Francisco Curbera, Frank Leymann, Tony Storey, and Donald F. Ferguson. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More.* Indiana: Pretince Hall, 2005.

WS4DSEC. *Reliable Secure Web Services for Devices.* 2010. http://www.ws4dsec.org (accessed January 15, 2011).

WS-I. "Basic Profile Version 1.2." *Web Services Interoperability Organization.* 2010. http://ws-i.org/Profiles/BasicProfile-1.2-2010-11-09.html (accessed November 20, 2010).

—. "Ensuring Web Services Security." *Web Services Interoperability Organization.* 2004. http://www.ws-i.org/resource/presentations.aspx (accessed August 12, 2010).

—. "Secure and Interoperable Web services." *Web Services Interoperability Organization.* 2007. http://www.ws-i.org/resource/presentations.aspx (accessed August 3, 2010).

Zeeb, Elmar, Andreas Bobek, Hendrik Bohn, and Frank Golatowski. "Service-Oriented Architectures for Embedded Systems Using Devices Profile for Web Services." *21st International Conference on Advanced Information Networking and Applications Workshops AINAW07.* Niagara Falls, 2007. 956-963.

Zeng, L, B Benatallah, M Dumas, J Kalagnanam, and Q Sheng. "Quality driven Web Services composition." *In Proceedings of the International WWW Conference 2003.* Hungary, 2003. 411-421.

Zhang, Yan, Jun Zheng, and Mio Ma. *Handbook of Research on Wireless Security.* London: IGI Global, 2008.

8 APPENDIX A - WS STANDARDS OVERVIEW

(INNOQ 2007)

Metadata Specifications

WS-Policy
1.5
W3C
Working Draft

▲ **WS-Policy** describes the capabilities and constraints of the policies on intermediaries and endpoints (e.g. business rules, required security tokens, supported encryption algorithms, privacy rules).

WS-PolicyAssertions
1.1
BEA Systems,
IBM, Microsoft, SAP
Public Draft

▲ **WS-PolicyAssertions** provides an initial set of assertions to address some common needs of Web Services applications.

WS-PolicyAttachment
1.2
W3C
W3C Member Submission

▲ **WS-PolicyAttachment** defines two general-purpose mechanisms for associating policies with the subjects to which they apply; the policies may be defined as part of existing metadata about the subject or the policies may be defined independently and associated through an external binding to the subject.

WS-Discovery
Microsoft, BEA Systems, Canon,
Intel and webMethods
Draft

▲ **WS-Discovery** defines a multicast discovery protocol for dynamic discovery of services on ad-hoc and managed networks.

WS-MetadataExchange
1.1
BEA Systems, Computer Associates,
IBM, Microsoft, SAP, Sun Microsystems and
webMethods
Public Draft

▲ **WS-MetadataExchange** enables a service to provide metadata to others through a Web services interface. Given only a reference to a Web service, a user can access a set of WSDL/SOAP operations to retrieve the metadata that describes the service.

Universal Description, Discovery and Integration (UDDI)
3.0.2
OASIS
OASIS-Standard

▲ **Universal Description, Discovery and Integration (UDDI)** defines a set of services supporting the description and discovery of businesses, organizations, and other Web services providers, the Web services they make available, and the technical interfaces which may be used to access those services.

Web Service Description Language 2.0 SOAP Binding
2.0
W3C · Working Draft

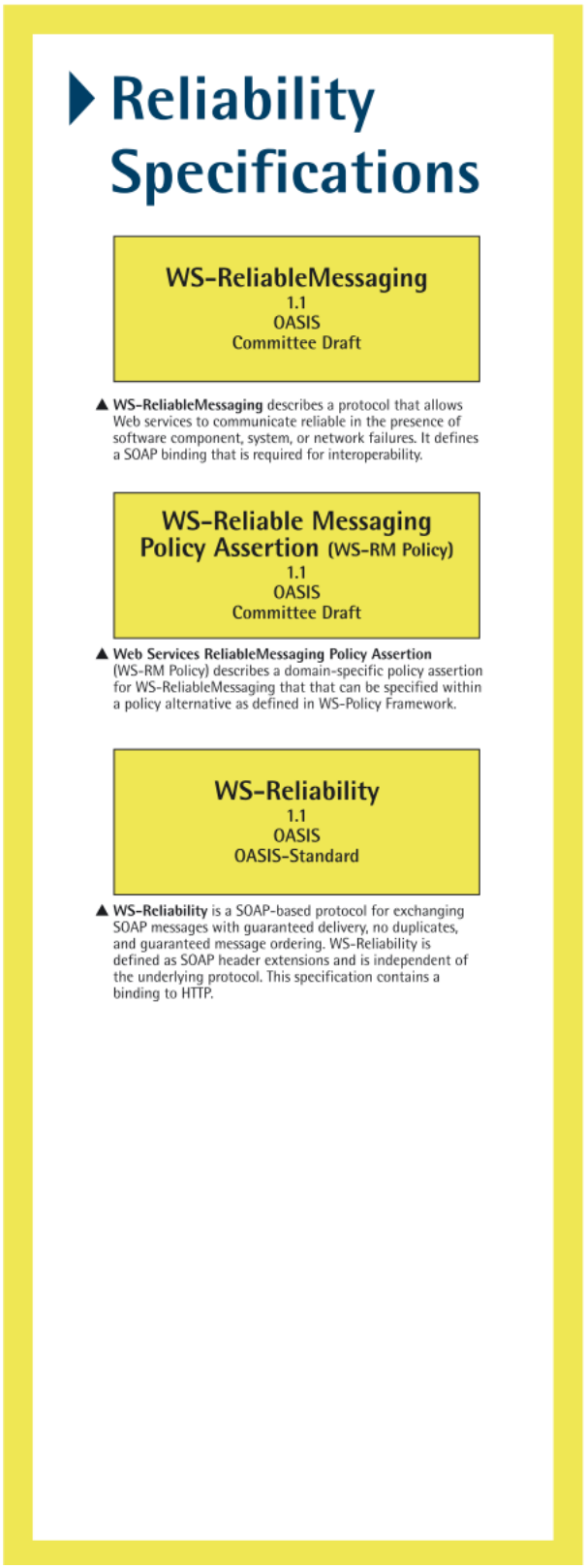
▲ **Web Service Description Language SOAP Binding** describes the concrete details for using WSDL 2.0 in conjunction with SOAP 1.1 protocol.

Web Service Description Language 2.0 Core
2.0
W3C
Candidate Recommendation

▲ **Web Service Description Language 2.0 Core** is an XML-based language for describing Web services and how to access them. It specifies the location of the service and the operations (or methods) the service exposes.

Web Service Description Language 1.1
1.1
W3C
Note

▲ **Web Service Description Language 1.1** is an XML-based language for describing Web services and how to access them. It specifies the location of the service and the operations (or methods) the service exposes.


Reliability Specifications
WS-ReliableMessaging
1.1
OASIS
Committee Draft
▲ WS-ReliableMessaging describes a protocol that allows Web services to communicate reliable in the presence of software component, system, or network failures. It defines a SOAP binding that is required for interoperability.
WS-Reliable Messaging Policy Assertion (WS-RM Policy)
1.1
OASIS
Committee Draft
▲ Web Services ReliableMessaging Policy Assertion (WS-RM Policy) describes a domain-specific policy assertion for WS-ReliableMessaging that that can be specified within a policy alternative as defined in WS-Policy Framework.
WS-Reliability
1.1
OASIS
OASIS-Standard
▲ WS-Reliability is a SOAP-based protocol for exchanging SOAP messages with guaranteed delivery, no duplicates, and guaranteed message ordering. WS-Reliability is defined as SOAP header extensions and is independent of the underlying protocol. This specification contains a binding to HTTP.

9 APPENDIX B – INTERACTION OF WS SPECIFICATIONS (SOASPECS 2009)