ARES project CONSOLIDER-INGENIO 2010 CSD2007-00004 Workpackage 1 - Task 4 (WP1.T4) Development of an Intelligent Authorization Gateway

Cristina Alcaraz[1], Gerardo Fernandez[1], Carmen Fernandez-Gago[1], Jordi Forne[2], Francisca Hinarejos[2], Javier Lopez[1], Pablo Najera[1], Rodrigo Roman[1]

[1] University of Malaga
Severo Ochoa 4, 29590 - Malaga (Spain)
{alcaraz,gerardo,mcgago,jlm,najera,roman}@lcc.uma.es
[2] Technical University of Catalonia
1-3 Jordi Girona, C3 08034 Barcelona (Spain)
{jforne}@entel.upc.edu, xisca.hinarejos@uib.es

November 3, 2011

Abstract. This report summarizes the advances carried out for task WP1.T4 within the ARES project to develop an Intelligent Authorization Gateway for Critical Infrastructures. The reports starts by explaining how various technologies, such as Wireless Sensor Networks, Ad-Hoc networks, and the Internet are beneficial for present-day SCADA (Supervisory Control and Data Acquisition) systems, highlighting the diverse security challenges that must be taken into account. After stating that authorization is one of those challenges, this report describes the design of an authorization gateway, which allows diverse actors (global/on-site operators, operators from different SCADA systems, "backbone" sensor nodes) to provide credentials in order to access various SCADA services. The design of the different elements of this authorization gateway, which come from results obtained from this and other tasks (WP1.T2, WP1.T3, WP2.T1), is also described. Finally, a list of published works and conclusions are also provided.

1 Introduction

In a critical infrastructure, providing authentication and access control services to the different actors that participate in the management of the different critical systems is one of the primary security challenges in the field [**?**]. This challenge becomes even more significant in heterogeneous

environments, where different technologies are used to connect, supervise and control the existing infrastructures. For example, a Wireless Sensor Network (WSN) can coexist with a traditional RTU-based (Remote Terminal Unit) system retrieving information from the different sensors, switches and pumps. Besides, it is possible to retrieve these readings using a wide range of communication technologies: from local wireless networks (on-site operators) to Internet-based connections with centralized SCADA (Supervisory Control and Data Acquisition) systems. Therefore, there is the need to develop specific authorization security services that can be used by all actors of the critical infrastructure, regardless of the existing underlying technologies.

Our main contribution in this report is the development of an authorization gateway for highly heterogeneous SCADA systems. SCADA systems have been commonly used to monitor large infrastructures, and very recently there have been major advances in this field, with new technologies being applied to reduce costs and improve their monitoring capabilities. For example, the Internet provides global connectivity independently of the physical locations of components/members of a system. Moreover, wireless communication reduces the overall costs while improving the mobility and connectivity with other control components. However, all these technologies have their own inherent security problems, which can affect the reliability of the SCADA system.

Our first objective was to study the potential benefits of interconnecting SCADA systems with other technologies, in order to prove that such interconnection is beneficial for critical infrastructures. This study not only included the advantages of using these technologies, but also the existing security problems and the mechanisms that should be used to limit their effect in the infrastructures. Precisely, one of the outputs of this work explicitly illustrated the need of developing an authorization gateway for this particular context. Therefore, our second objective was to develop the functional architecture of such authorization gateway. As a result, we developed three functional components that can be used to control the access of all involved actors: from SCADA and on-site operators to proactive ("backbone") sensor nodes. The authorization architecture is also linked with the pervasive authentication system developed in **WP2.T1**. The design behind one of the functional components is fully explained in this report, while the other two functional components, which are based on results obtained in previous tasks (**WP1.T2** and **WP1.T3**), will be overviewed in diverse appendices.

The rest of the report is organized as follows: Section 2 presents the benefits and security challenges of interconnecting SCADA systems with Wireless Networks and Internet technologies. After highlighting the problem of distributed authorization, we will overview the authorization gateway in Section 3, alongside with the mechanisms used to connect "backbone sensors" to the Internet. Finally, we will conclude the report and show the papers that corroborate its contents in Section 4, and in Appendix A and Appendix B we will review the design behind various elements of the authorization gateway that have been considered in other deliverables.

2 Interconnected SCADA Systems

As it has been mentioned in previous tasks in this same workpackage, SCADA (Supervisory Control and Data Acquisition) systems are located at the heart of the critical infrastructures. Their main purpose is to supervise and control the behaviour of the different elements of a critical infrastructure, such as the boiling water reactor of a nuclear power plant. SCADA systems have emerged from their isolation, as not only they can collaborate with each other in order to monitor the state and dependencies of multiple infrastructures, but also can integrate different technologies such as Wireless Sensor Networks (WSN), Mobile Ad-hoc Networks (MANETs), and the Internet. Still, due to the specific characteristics of these technologies, this integration can lead to multiple security challenges.

The purpose of this section is twofold. First, we want to highlight the benefits of interconnecting SCADA systems with Wireless Networks and Internet technologies, in order to prove that these technologies are essential for present-day critical infrastructures and should be considered as part of the authorization gateway use case. Second, we want to present the main security challenges that appear due to this integration. Not only this security analysis can help to pinpoint the major issues that the infrastructure designers must take into account so as to strengthen the robustness and reliability of their systems, but also explicitly mentions that there is a need for a authorization service (that is, our authorization gateway).

2.1 SCADA Technologies

2.1.1 Networked SCADA systems

Industrial Control Systems (ICS) are complex systems in charge of performing a set of specific control tasks, becoming an essential part of an industrial production process. They are considered the main framework of supervision and monitoring of other critical infrastructures, such as: electric energy systems, nuclear energy systems, water and sewage treatment plants, gas/oil energy systems or transportation systems. In particular, they are able to remotely monitor and supervise engineering devices installed or deployed close to the critical infrastructure, manage automation and operational tasks, and store sensitive information. This type of information can be both data measurements, i.e. physical events relative to real conditions of controlled infrastructures, and alarms, i.e. control messages labelled with priority levels.

In the existing literature, there are two types of ICSs [**?**]. They can be distinguished from each other by their dimensions and geographical distributions. Namely,

- *Supervisory Control and Data Acquisition* (SCADA) System. A SCADA system is a distributed network over large geographic areas where a set of industrial automation services are offered to control the performance and continuity of other critical infrastructures, such as: electric energy systems, nuclear energy systems, water and sewage treatment plants or transportation systems.
- *Distributed Control Systems* (DCS). These systems have the same functionality as a SCADA system but they are geographically closer to manufacturing operations and industrial facilities. It is very important to highlight that throughout this report, we will use the term SCADA to cover any monitoring and control procedure for both SCADA systems and DCS systems.

Since SCADA Systems were first introduced in the 1960s, three main generations have been emerged: *Monolithic, Distributed and Networked*, all of which share a number of characteristics. Firstly, they have adopted the existing ICTs in order to improve the monitoring processes in real-time, as well as the performance and availability of controlled infrastructure (e.g. large industrial lines of oil pipelines). Secondly, they share three types of sub-networks: (i) the central network, (ii) remote substations and (iii) the corporative network. The operations carried out in the central network are related to the control and management of the critical

infrastructures. Such operations are managed through specific operator consoles or human-machine interfaces (HMIs), which allow operators to read specific physical parameters (e.g. pressure, electrical signals, temperature, etc) or alarms received from remote substations, or even transmit certain commands (e.g. open/close pumps) to specific field devices localized in remote substations. On the other hand, the operations carried out in the corporative network are directly related to the general supervision of the system whose accesses databases and servers installed in the central network are rather more restricted.

The first SCADA networks were designed in the *Monolithic* generation under a centralized control in a mainframe system. This mainframe was configured as the primary control system; while another mainframe system was configured as the standby in order to cover any functionality of the system in the event of a failure in the main system. The architecture of a substation was basically based on one or several special remote terminal units (RTUs), which had limited memory and processing capabilities with output/input (O/I) interfaces to measure/actuate physical signals. These signals had to be retransmitted to the central system via telephone or radio with a low data transmission rate and through property automation protocols such as for instance Modbus serial or IEC-101. Later, in the second *Distributed* generation of SCADA systems, the monitoring processes were distributed among different network components, with multiple master terminal units (MTUs) that established communication with the substations. In fact, the communication with remote substations was established using large (distributed or hierarchical) local-area networks controlled by these MTUs. The RTUs, configured in such substations, were equipped with advanced serial I/O interfaces with faster microprocessors, memory and math coprocessors to support complex applications.

Finally, the latest advances in SCADA systems are seen in the third generation or *Networked* generation. This generation broke with the isolation concept of the previous generations by including in its network designs open connections using the TCP/IP (Transmission Control Protocol/Internet Protocol). All these technical advances also came to substations, where RTUs were able to provide hierarchical and inter-RTU communication (i.e., interconnectivity among RTUs) under TCP/IP. This fact helped RTUs work as data concentrators to store large data streams or as remote access controllers to autonomously and remotely reconfigure/recover parts of the system. Moreover, Wireless technologies deployed on the substation alongside with RTUs provide mobility and local control

with a low installation and maintenance cost. In fact, wireless communication has been recently standardized to ensure the reliability in the communications and constant performance, with standards such as Zigbee PRO, WirelessHART and ISA100.11a. Both the impact of the Internet and the different Wireless technologies will be further explained in the next paragraphs.

2.1.2 SCADA Systems and the Internet

As was previously commented, the Internet is one of the most demanded technologies by Industry. This special interest is due to the fact that the Internet provides global connectivity independently of the physical locations of components/members of a system. Its public communication infrastructure offers Web solutions, as well as flexibility in the data acquisition and management, data dissemination, maintenance, diagnosis, and interfaces to visualize data streams and resources in real time. In addition, the use of open standards and open Web protocols (e.g. HTML, HTTP or HTTPS) can also significantly reduce costs in terms of hardware and software, time, personnel and field operations [**?**]. As a result, researchers, engineers and commercial companies are jointly working to study the impact of using the Internet and Web solutions in critical control systems. For instance, Qui et al. proposed in [**?**] a WSDS (Web-based SCADA Display Systems) system to access the system through the Internet. The same authors also proposed a Web-based SCADA display system based on very-large-scale integration (VSLI) information technologies in [**?**]. Similarly, Leou et al. proposed in [**?**] a database management system to centralize the critical data received, providing a Web-based power quality monitoring system. Li et al. presented in [**?**] a Web-based system for intelligent RTUs with capability for interpreting HTTP. Jain et al. presented in [**?**] a Web-based expert system for diagnosis and control of power systems. Lastly, several commercial companies, such as for instance Yokogawa [**?**] or WebSCADA [**?**], have some Web control solutions already available for the market.

Nevertheless, the use of the Internet could give rise to new security threats and reliability problems in the system. Examples of attacks may be intercepted communication channels, disruption of services, isolation or data alteration. One way of protecting the communication channels could be to use SSL (Secure Sockets Layer)/TLS (Transport Layer Security) services offered by the TCP/IP standard, hard cryptographic primitives, hash functions, key management systems and intelligent mechanisms, such as Intrusion Detection Systems (IDS), Firewalls or Virtual

Private Networks (VPNs). This last security mechanism may even be considered a cost-effective high speed communication solution between substations and the SCADA central network over a shared network infrastructure, while simultaneously providing both the functionalities and the benefits of a dedicated private infrastructure [**?**]. On the other hand, it is also necessary to configure authentication mechanisms to verify the authorized access resources and services in the system, as well as authorization mechanisms to prove an entity's identity and rights in the management of critical data and commands. Data redundancy mechanisms should also be installed to ensure the data availability at any time and from anywhere, as well as registering incidents or anomalous events occurring. Security policies should be put in place and frequent training courses should be available to users to avoid unintentional actions.

2.1.3 Wireless Networks and SCADA Systems

Another essential technology in automation and control processes is wireless communication, for several important reasons. This technology is able to provide (1) control as a wired infrastructure but with a low installation and maintenance cost, (2) mobility and (3) connectivity with other control components independently of the environmental conditions. To be more precise, many of the critical infrastructures control conditions which are impossible for humans to monitor in person (e.g., high/low temperatures, high/low pressures, noise, underground water/oil pipelines, etc.). These critical conditions force systems to deploy autonomous and intelligent devices in order to cover certain functionalities in these areas (e.g., robots, automation vehicles, sensors, active RFID devices, etc.). In fact, the vast majority of wireless technologies have already been proposed to be included in industrial control networks, such as Bluetooth, WiFi, Mobile technology (UMTS, GPRS or TETRA), Satellite, Global Positioning System (GPS), WiMAX, microwave, Mobile Ad-hoc Networks (MANETs), or Wireless Sensor Networks (WSNs).

MANETs are becoming increasingly important for the protection of critical infrastructure, since these can provide an alternative communication link between operators and the remote substations. Thus operators localized close to the substation can locally and directly manage the information retrieved from sensors in the substation without requiring to go through the SCADA centre while offering mobility in the area. On the other hand, this network is self-organizing and can be easily deployed without any infrastructure while allowing operators to access determined points in real-time to attend critical alarms, in addition to facilitate them

a quick reaction and reconfiguration of a part of the system by receiving a high level of incidences. Note that the control in these types of networks could be limited to a determined distance range and a number of nodes.

Furthermore, WSNs can offer control as an RTU while ensuring prevention of abnormal situations thanks to their sensor nodes, which are equipped with a 4MHz-32MHz micro-processor, 8KB-128KB RAM, and 128KB-192KB ROM. Generally, and depending on the application context, the nodes are linked to an energy supplier or to industrial equipment in order to maximize their lifetime (by between 5 and 10 years). Their sensor nodes, smart and autonomous devices, are capable of processing any information sensed from their sensors and transmitting it to a central system with considerable hardware and software resources, such as for example an RTU working either as a data collection device.

As was previously mentioned, some international organisations such as ZigBee Alliance, HART and ISA, have taken advantage of these aspects to specify and launch new wireless communication standards for WSNs under a purely industrial contexts, such as ZigBee PRO [**?**], WirelessHART and ISA100.11a. These specifications have several common objectives, such as energy saving, coexistence with other communication systems, communication reliability and security. However, due to the critical nature of the application context, the nature of wireless networks, which tend to be generally susceptible to attacks, and the vulnerable nature of the technology used, it is necessary to ensure security and reliability in wireless monitoring processes. For example, the security in WSNs is mainly supported by Symmetric Key Cryptography (SKC) primitives because of the high hardware and software constraints of the sensor nodes.

2.2 SCADA Interconnection: Benefits and Challenges

This section describes the benefits of integrating SCADA systems with the technologies described in the previous section, showing a framework where sensors nodes, MANETs and the Internet play an important role. In fact these pieces allow the creation of a distributed preventive mechanism for the protection of industrial systems. Moreover, we will highlight the major security challenges of this integration, showing that authorization has not only a primary role in the development of the framework but also is underdeveloped and needs of a intelligent authorization gateway.

2.2.1 Framework: Integration

Considering the technologies discussed in sections 2.1.2 and 2.1.3, the next step is to analyze how to combine them in the same context without

Technologies	*Management*	*Mobility*	*Collaboration*	*Detection*	*Alert*	*Response*
WSN	✓ $_{RTU+WSN}$	-	-	✓	✓	-
MANET	✓ $_{RTU}$	✓	✓ $_{Local}$	-	-	-
Internet	✓ $_{RTU}$	-	✓ $_{Global}$	-	-	-
WSN-MANET	✓ $_{RTU+WSN}$	✓	✓ $_{Local}$	✓	✓	✓
WSN-Internet	✓ $_{RTU+WSN}$	-	✓ $_{Global}$	✓	✓	✓
MANET - Internet	✓ $_{RTU}$	✓	✓ $_{Global}$	-	-	-
WSN-MANET-Internet	✓ $_{RTU+WSN}$	✓	✓ $_{Global}$	✓	✓	✓

Table 1: Analysis of Properties using WNSs, MANETs and the Internet

risking the business continuity. The result of this action will be a framework whose main objective is to assure a full coverage in the monitoring processes by taking advantage of the benefits of using the Internet and mobile ad-hoc networks. Besides, such framework has to be able to detect anomalous phenomenons and faults in the system, to constantly register the interactions among different elements and events, and to alert about failures and threats as soon as possible to respond in time.

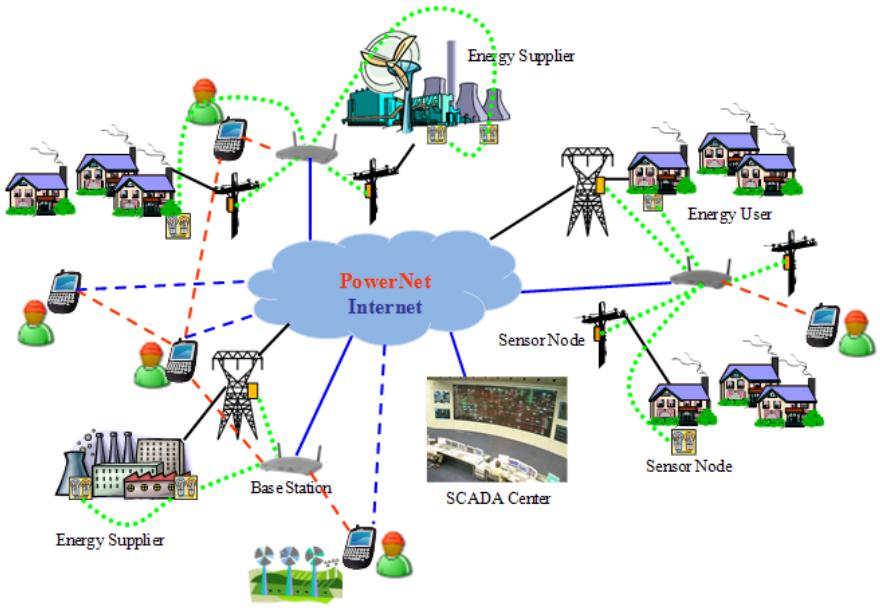


Fig. 1: Integration of WSNs, MANETs and the Internet in a same Context.

To understand in detail the design of this approach, a set of properties (summarized in the Table 1) have to be discussed. These properties are associated to the management, mobility, collaboration, detection, alert and response. The management is referenced to an operator's action to manage a substation (through an RTU/the station base of a WSN) with commands and queries. Mobility is an operator's ability to move without losing the control from a substation. Collaboration is an operator's capability for interacting with another components (e.g. MANETs devices, sensor nodes, the SCADA centre). Both detection, alert and response are the ability of the system to prevent an anomalous events. In particular, the response is associated to an operator's capability to react to anomalous events. As can be noted in the table, each technology offers by themselves a set of properties useful for our approach. However the individual use of these technologies do not help achieve the requirements of our approach: control and prevention (an essential parameter for the protection). For instance, an isolated WSN in a substation can offer both management and detection but no response. To avoid this, it is necessary to use all of these technologies in a same context.

For the sake of clarity, the Figure 1 depicts an overview of how different communication technologies can interact with each other in a same industrial context. Here, the sensor nodes are deployed in the whole system, from energy suppliers to energy users for the data acquisition. These smart devices allow the system to obtain the measurements regarding energy generation, distribution and consumption, which are sent back to the SCADA center to remotely control and manage the energy distribution. In addition, as existing standards for WSNs such as ZigBee, WirelessHART and ISA100.11a provide some services that allow the coexistence with other technologies, it is possible for operators in the field equipped with diverse mobile communication devices (such as a PDA, a laptop, a cell phone, and any other lightweight, easily transportable computing device) to interact with them. In fact, the connectivity WSN-MANET facilitates operators to locally manage data streams and locate a problem detected by sensors. Furthermore, MANETs allow the creation of collaborative links with other operators in order to respond immediately during extreme situations (like for example, a circuit break).

Regarding to the Internet, as we show in the Figure 2, it glues some of the previous technologies and elements (WSNs, MANETs and the SCADA Centre) together. A WSN deployed in a remote substation can be managed by distant operators or by a SCADA centre, who receive queries in real-time, carry out control processes, and manage anomalous

behaviors in real-time. Furthermore, the sensor nodes of the WSN could even access the Internet by themselves, generating alerts (due to either a failure or a threat) in order to prevent situations that may damage the normal performance of the services. As a result, the Internet help the system maximize its collaboration capabilities, offering management from anywhere at any time, as well as a timely response to assure reliability and continuity of services.

The data acquisition world (WSNs) and all other elements (e.g. the SCADA centre) are usually connected through an interface known as the base station. The role of such base station will depend on both the communication standard used and on the requirements associated to the network topology. For example, ZigBee PRO is based on a mesh/star network that contains a coordinator (a trust node in charge of managing the deployment, maintenance and control processes), routers (to help devices to transmit data to the coordinator), and sensor nodes. On the other hand, WirelessHART uses gateways, a network manager (a trust node that could be integrated into the gateway), sensor nodes, and the existing industrial devices (e.g. an RTU). The network manager is a device with enough resources to manage the routing tables, the synchronization schedule, the network configuration and the security in the whole network. ISA.100.11a provides a network architecture that is similar to WirelessHART, but using i) backbone nodes that directly connect the Internet and ii) two specific managers that can be integrated in the gateway, one of them in charge of managing resources and communication, and the other one in charge of providing security. In any case, it is very important to identify the role of the base station. In the case of a ZigBee PRO network, it would be represented by the coordinator, whereas in a WirelessHART and ISA100.11.a network would be represented by the managers.

Another aspect that must be taken into account is the connectivity model that links the WSNs with external networks such as the Internet. Currently, three Internet connectivity models have been identified in [**?**]: *Front-End* Proxy solution (where WSNs are completely independent from the Internet), *Gateway* solution (where WSNs retain their protocol independence but are able to exchange information directly with Internet hosts) and *TCP/IP* solution (where WSNs implement a TCP/IP-compliant stack). In order to find the most adequate solution for our critical application context, it is necessary to consider the complexity and effects of these solutions. In our particular case, the system must access data streams and issue control commands, tasks that can be provided

by the Front-End solution. Such solution will centralize all the computational overhead in a set of base stations, whose task will be to parse and store any type of information (allowing the existence of an historic repository and store-and-forward strategies), and interpret and translate control SCADA commands to a protocol that the sensor nodes can understand. On the other hand, in the TCP/IP solution and in the Gateway solution, the external entities will have to interact directly with the sensor nodes. This can be problematic in a critical context, mainly due to the capabilities of the nodes: the sensor nodes will be specially vulnerable against attacks launched by remote entities, and there will be an extra overhead caused by the management of the protocols and the security services. Nevertheless, note that there can exist special nodes (i.e. a "backbone" of nodes) that can connect to the Internet by themselves, sending special messages such as alarm messages.

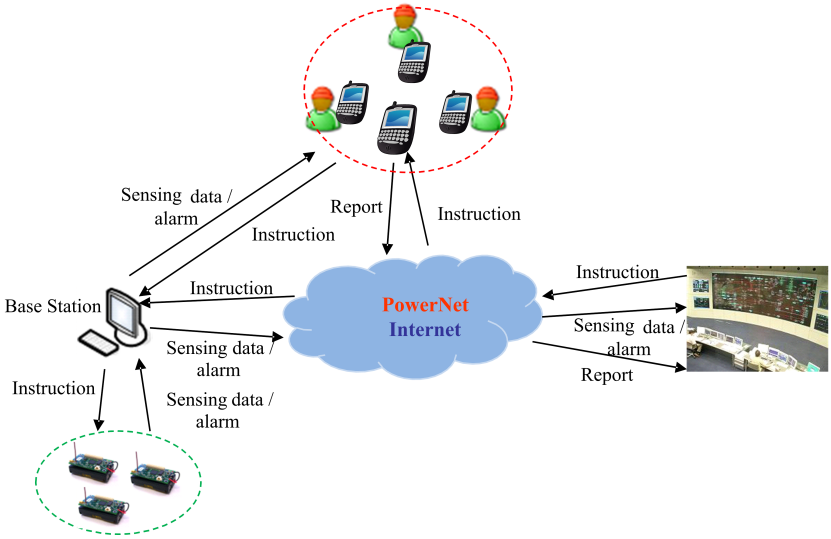


Fig. 2: Conglomeration of Entities and their Communications

2.2.2 Framework: Security Challenges

As for the security of the previously presented interconnection framework, it needs to be carefully considered, as data streams have to be transmitted among different types of networks and processed on different types of

devices with very different capabilities. In a particular case, an attacker may, for instance, target some link to isolate determined system parts. To avoid this fact, cryptographic primitives and protocols need to be designed carefully, and the block size of ciphertext and the packet size need to be optimized according to the message type in order to minimize unnecessary padding. The impact is especially important for WSNs so that these will reduce the communication overheads, and thus saving the restricted bandwidth, reducing the traffic jam and packet loss, in addition to save energy consumption to extend their lifetime.

Other security aspects to take into account in critical systems are availability, detection and accountability. Availability allows a system to get services and data streams independent of the real state of a node (e.g. without connectivity or energy) in the network. To this end, it is necessary to implement mechanisms and protocols that provide data redundancy, or manage backup copies in high resource systems, like a base station. Likewise, detection and accountability are achieved by implementing specific incident management mechanisms in charge of registering the sequences of events that happen within the system. Such functionality can be mainly performed by the base station, which acts as the link point among different communication networks. It will analyze the information traffic and the internal behavior of each network, registering anomalies and/or events occurred.

The system must also guarantee reliability in the communications to satisfy the critical infrastructure protection standard [**?**]. This requires that the sensing data streams and alarms should reach the SCADA control center securely, reliably and timely. Attacks that target part of the communication channels spanning different network domains should not stop the transmission of critical information. This dependency implicates that backup communication paths should be available when the normal path is under attack. In addition, it is also important to take into account some security issues related to the integration of a WSN to the Internet [**?**]. For example, in the particular case of a Front-End solution, an adversary could take advantage of the centralized nature of the base station in order to disrupt its functionality. A way of mitigating this fact would be to increase the number of base stations (redundant systems) so as to improve their availability. Also, a base station has to be configured with security mechanisms, which must protect the information flow between it and an Internet host under a formal security standard, and between it and the sensor nodes under a security approach defined for WSNs. Finally, the sensor nodes themselves must be able to manage any type of

incidence, guaranteeing a timely response, and allowing reconfiguration and self-healing.

Finally, Authentication and access control across different network domains is another major issue. It should consider the complicated topologies of different networks and the different roles of mobile devices, and authentication issue and access control in a unified way in pervasive mobile networking environment. Following on this same line, the authorization should also be considered to prove an entity's identity and rights to manage measurements, alarms or instructions. Precisely, this role can be covered by the authorization gateway, presented in the next section.

3 Authorization Gateway for SCADA Systems

After the benefits and security challenges of integrating SCADA systems with WSNs, MANETs and the Internet have been discussed, this section presents the functional architecture of the security service that tackles the problem of authentication and access control in this particular context: the Authorization Gateway.

3.1 Overall Structure

3.1.1 Context and Requirements

The main goal of the authorization gateway is to check whether the users of a critical infrastructure can access to a particular service. Note that this gateway must be able to take into account all the possible users and services, regardless of their location. Consequently, it is necessary to describe in detail not only the different actors that will make use of the gateway, but also the context where the gateway will run.

The following actors will make use of the authorization gateway:

1. *Global Operators.* These operators are in charge of obtaining information from the different elements of the Critical Infrastructures, such as the RTUs, the Wireless Sensor Nodes, and other SCADA systems. Note that these operators are located within the physical premises of the SCADA system.
2. *On-site Operators.* These operators make use of a MANET inside the substation in order to get information from the different sensors and actuators.
3. *Sensor Nodes.* In some cases the sensor nodes (or "backbone nodes" as mentioned in section 2.2.1) can be able to send information directly to certain SCADA operators or alert management systems.

As for the context of the gateway, it was described in Figure 2, and will be summarized here. A network of *sensors* (including wireless sensors and RTUs) provide their services through a *base station* located in the substation. *On-site operators* can interact directly with the base station or with the SCADA systems, while *global operators* manage all infrastructures through the Human-Computer Interfaces (HCI) of the SCADA systems. All entities make use of the Internet to communicate with each other, with the exception of the On-site operators that can access the base stations directly if needed.

Finally, the main requirements for the authentication gateway are as follows:

- *Correctness.* The authorization gateway must help all relevant actors to open a secure communication channel with each other if they are authorized to do so.
- *Robustness.* The authorization gateway must not degrade the robustness of the critical infrastructure, becoming a single point of failure.
- *Flexibility.* The Authorization gateway must be able to provide its services to different heterogeneous actors.

3.1.2 Architecture

The authorization gateway is composed mainly by the following three functional components:

1. *Substation component*: This component allows global/on-site operators to access the information coming from the substations managed by their SCADA system.
2. *Collaboration component*: This component allows operators from different SCADA systems to access the information provided by all substations.
3. *Sensor component*: This component is used by "backbone" sensor nodes to send information beyond the base station.

The *substation component* stores the credentials and permissions from the different operators. Both global and on-site operators access the authorization gateway and provide proof of their identity using various authentication mechanisms. Afterwards, the substation component checks if the operator is authorized to perform a certain operation and provides an authorization token. This token is later used to access the services provided by the sensor nodes / RTUs. This component can deployed in a centralized or in a distributed way. If centralized, operators will access

the SCADA system that controls that particular substation and will interact with it using the authorization token. If distributed, the substations themselves will grant access to their services, thus the authorization token might not be needed (i.e. substations directly open a secure channel to the service provider once the authorization is performed). Note that, as shown in the diverse articles from **WP1.T2, "Design of authentication, authorization and logging services specifically for the CII"** (cf. Appendix A), the sensor nodes and the RTUs provide their services through the Base Station(s). This simplifies the process of managing the security credentials, as every individual sensor node only needs to trust the base station that controls it.

As for the *collaboration component*, its services are based on attribute delegation, which has been studied in detail in the articles from **WP1.T3, "Development of delegation services for distributed authorization"** (cf. Appendix B). In short, an operator sends a request to the service provider, requesting the attributes that are needed to make use of the service. After retrieving the attributes, the operator contacts an attribute federation system (made by the different collaboration components located in the different SCADA systems), and if it is authorized, retrieves an attribute certificate. This certificate is then sent to the service provider, which will check if it is signed by its collaboration component. If everything is correct, the operator will ba able to access the services. This approach is both centralized and distributed: an Attribute Authority (AA) defines the Global Attributes that are known by all SCADA systems, while Attribute Ontology Servers (AOS) are located in every SCADA system and define those Domain Attributes whose meaning is restricted to the domain in which they were defined. Note that during the authorization process all actors only need to contact those AOS that manage the attributes of their respective SCADA systems, as the AA can define the relationship between the different global and domain attributes in an offline fashion.

Finally, the *sensor component* is in charge of maintaining and distributing the credentials that will be used by backbone sensor nodes in order to send specific information (e.g. alarms) beyond the boundaries of the base station. These credentials, which will be used to open a secure channel between the sensor nodes and the external entities, function as an implicit authorization token. However, it is necessary to analyze which kind of credentials can be used in this particular case: not only sensor nodes in industrial environments are limited in terms of computational power, but also the speed of the wireless channel (e.g. IEEE 802.15.4) can

be very limited. This analysis has been carried out in Section 3.2. The conclusion is that constrained nodes can make use of Public Key Cryptography (PKC) if a slight delay (i.e. around 2-3 seconds) is acceptable. This way, both backbone sensors and external entities can authenticate each other using certificates signed by the Authorization Gateway. Note that in order for this approach to work inbound connections (i.e. direct channels from the external entities to the backbone sensor nodes) must be prohibited, as limited sensor nodes must not work as web servers if they use PKC as an authentication mechanism. Summarizing: backbone sensor nodes can open a direct communication channel with an external entity, but these external entities must make use of the substation component and the base stations if they want to access the services of the nodes. Finally, if this slight delay on the processing of the alarm is unacceptable, the backbone nodes should become powerful nodes able to execute PKC in less than 0.1 seconds.

Requirement Analysis. After presenting the different components of the authentication gateway, we have to check if they fulfill the requirements presented in Section 3.1.1. Such requirements are satisfied, as shown below:

- *Correctness.* The Authorization Gateway allows all actors to access the services of the critical infrastructure:
  1. *Global and On-site Operators.* All operators can use the functionality of the *substation component* to access to the services provided by their particular substation. If an operator wants to access a different substation, they can use the *collaboration component* to obtain the certificates that allow them to complete their task.
  2. *Sensor Nodes.* Backbone nodes will use the certificates provided by the *collaboration component* to access external entities.
- *Robustness.* All authorization gateway components can be deployed in a distributed fashion where needed. Besides, they can be protected using different network protection mechanisms (e.g. firewalls, redundant servers).
- *Flexibility.* As shown by the *correctness* requirement, the different components of the authorization gateway allow it to provides its services to a heterogeneous set of actors. Beyond that, the different components of the authorization gateway can be deployed in different ways (e.g. the *substation component* can be located in the SCADA system or distributed in the substations that are controlled by that SCADA system).

3.1.3 Mobile Authorization

The authorization gateway presented in this section can also interact with the pervasive authentication infrastructure [**?**,**?**] developed in **WP2.T1, "Study, proposal and software implementation of authentication/authorization schemes in ubiquitous scenarios"**. The goal is to allow operators to manage certain elements of the critical infrastructure even if the authorization gateway is not available. A summary of this authentication infrastructure is explained in the following paragraph.

The foundation of the pervasive authentication infrastructure is a pervasive public key infrastructure (pervasive-PKI). This infrastructure is able to provide authentication and access control services for users roaming between different heterogeneous networks, which can be very beneficial for operators working in the field. In this sense, the pervasive-PKI fully supports nomadic mobility, enabling secure services for users connecting through many different networking technologies (Wi-fi, UMTS, Bluetooth, etc.), and in multiple network topologies, even when global connectivity is lost and some services are temporarily unreachable. In fact, there are two modes of operation: connected mode (online trusted servers are available and traditional techniques are applicable for validation of user credentials) and disconnected mode (where different solutions, such as a new trust model and a collaborative model to obtain unavailable information, are implemented).

From an technological point of view, mobile users obtain their credentials from a X.509 authentication and authorization infrastructure (AAI, such as the Authorization Gateway) and they store the certificates in their devices or smart cards. Further authentication and authorization will imply the validation of these credentials. In connected mode, part of the AAI can be used to help credential validation, whereas in disconnected mode the infrastructure is not reachable, thus several cooperating software components installed in the user device have to be used: the PTM component (manages trust information about users, validates public-key certificates (PKCs), signs messages and verifies digital signatures), the Privilege Verifier (manages the validation of Attribute Certificates) and the Access Control Engine (in charge of decision-making based on the configured policies when controlling access to resources).

3.2 Connections from the Data Acquisition Network

In this subsection, we will explain in detail why PKC is a good option for creating a secure and authenticated channel between backbone nodes and

external entities. First, we will introduce the protocols that can be used to open such channel. Afterwards, we will introduce our assumptions and methodology, followed by the analysis. Finally, we will present the results of the analysis.

3.2.1 Protocols and Key Management

In our context, backbone sensor nodes (or simply sensor nodes) can directly interact with external entities (or external hosts) using Internet protocols, thus we should examine the existing security mechanisms and protocols used on the Internet that allow the creation of an authenticated secure channel. For example, with TLS, two peers can negotiate a common key for ensuring secure end-to-end transit at the transport layer. Also, at the web services layer, WS-Security and its associated specifications can be used to set up a secure channel in the general Web services framework, where languages and protocols such as WSDL, UDDI, and SOAP are used.

In TLS, a client and a server negotiate a stateful connection by using a handshaking procedure. This procedure is as follows [**?**]. First, both peers exchange hello messages to agree on the ciphersuite (i.e. cryptographic algorithms) to be used, exchange random values, and check for session resumption. Later, they exchange certificates and cryptographic information to authenticate themselves and to provide session information, which will be used to generate the shared secret. Finally, the client and the server verify that the handshake has finished correctly without tampering by an attacker. Most of the ciphersuites require of Public Key Cryptography (PKC) in order to authenticate the server (and the client) and to derive the shared secret, although the negotiation can make use of a pre-shared key by using the TLS-PSK ciphersuite [**?**]. Note that TLS is designed to be used with reliable transport channels such as TCP, but a datagram-compatible variant is also available [**?**].

As for web services security, the WS-Trust specification provides a framework for requesting and issuing security tokens, allowing applications to construct trusted SOAP message exchanges. This functionality is used by WS-SecureConversation [**?**], allowing parties to exchange multiple messages in a secure fashion for the lifetime of a communications session. There are three different ways for establishing the security tokens: a) created by a trusted third party known as security token service (STS), b) created by one of the agents and propagated with a message, and c) created by negotiation. In fact, the negotiation extensions allow both re-

questors and recipients to perform a set of exchanges prior to returning a security token.

Both TLS and the web services security protocols provide mechanisms for negotiating a shared secret between two peers. Most of these negotiation mechanisms are based either on Public Key Cryptography or on the existence of pre-shared master secret keys. By using PKC, two devices can exchange random values in a secure way (e.g. through PKC encryption) in order to generate a session key. As for pre-shared keys, these keys can be shared in advance amongst the communication parties, who will use them to create the session key. However, as PKC is supported but quite time consuming in constrained sensor nodes [**?**], and the use of a single pre-shared key might be unsuitable for certain IoT applications, it is necessary to analyse whether these key management mechanisms can be directly applied to Internet-enabled sensor nodes. It is important to point out that there are existing works that partially discuss this particular issue [**?**], and any reader interested in this area should also consider them.

3.2.2 Assumptions and Methodology

As our main goal is to analyse if Public Key Cryptography and pre-shared key strategies can be used to create secure channel between remote (constrained) entities, it is necessary to explain our assumptions and to state our methodology. Regarding the assumptions about the network structure and its elements, they are the following:

- *Accessing the WSN.* The routers that connect a WSN with the rest of the Internet must be able to convert any packets from IPv6 (Internet) to 6LoWPAN (WSN) and viceversa, as packets coming from both networks cannot be directly mapped. In this analysis we assume that such routers are in fact the base stations from the WSN, thus they can perform monitoring and management functions in certain situations.
- *Network structure.* In this analysis we will make use of the most "traditional" type of sensor network: a group of sensor nodes scattered in an industrial setting using IEEE 802.15.4 transceivers and connected to the Internet through a set of routers / base stations.
- *Sensor node features.* There are different classes of sensor nodes, ranging from "Class I" nodes with roughly 1kB of RAM and 16kB of instruction memory (e.g. Arduino Duemilanove), to powerful "Class III" nodes with 256kB of RAM and 4MB of instruction memory (e.g. Sun SPOT). In this analysis we will focus on the most common "Class

II" node (e.g. MICAz, IRIS). This type of node has 4-16kB of RAM and 48-256kB of instruction memory, and are equipped with a 4Mhz-8Mhz microcontroller [**?**]. Note that Industrial sensor nodes are actually quite similar to "Class II" nodes, but slightly more powerful. For example, ISA 100.11a-ready sensor nodes [**?**] can have 96kB RAM (containing both instructions and data), 128kB serial flash memory, 26MHz microcontrollers, and 80kB ROM.

- *Sensor node behaviour.* In this analysis we will assume that sensor nodes behave as web clients (client nodes), opening a direct connection with external entities. However, for the sake of completion, we will also assume that sensor nodes can behave as servers (server nodes), implementing web servers and providing services (even if sensor nodes should not be accessed directly in an industrial setting, as aforementioned). Therefore, sensor nodes can be regularly contacted by external entities ('pull'), but also can send information to external servers at predefined intervals ('push'). In our analysis, whenever we mention "clients" and "servers" we will refer to both sensor nodes and Internet hosts, while "client nodes" and "server nodes" refer only to sensor nodes.
- *Service model.* Although a sensor node can behave either as a client or as a server, in our analysis we will only consider the existence of the client/server model. That is, we will not consider the existence of P2P applications where multiple sensors and Internet hosts behave as clients and servers. Besides, we will not consider the existence of trusted third parties that could assist on the establishment of a secure channel.

Regarding our methodology, we will follow a property-oriented analysis. Every key management system has some properties, which are discussed below, that are relevant for the creation of a session key between remote entities. We will use these properties to compare the different key management systems and their suitability for the connection between data acquisition network sensors and Internet hosts.

- **Distribution.** This property is related to how the information that is needed for the negotiation of a shared secret (e.g. public key certificates, pre-shared keys) is distributed. Distribution is considered as *offline* if all the information is stored beforehand within the peers. A distribution can also be considered as *online* if part of the information can be distributed during the negotiation process. An online distribution is desirable, as any client would be able to establish a

secure channel with any server anytime, even if they did not meet before.

- **Authentication.** This property defines whether the peers (*client* and/or *servers*) are authenticated during the negotiation process or not. This can be achieved, for example, using of a uniquely shared key or a PKC signature. Note that in the negotiation process it is possible to authenticate a peer (e.g. if a pre-shared key is shared only with that peer) or a group (e.g. if a pre-shared key is shared with a group of peers). In our context, client authentication is important, server authentication is desirable, in order to assure clients that they are obtaining data from the right source.
- **Overhead.** This property specifies both the communication and computational overhead of executing the negotiation process. This property will refer mainly to the overhead imposed in sensor nodes, as i) they are much more resource constrained than other Internet hosts and ii) the energy consumed by sending and receiving information through the wireless channel is quite high, and most sensor nodes will be battery-powered. As expected, the overhead in our context must be as low as possible.
- **Resilience.** This property indicates the ability to cope with stolen credentials. The resilience of a key management system is low if an attacker that extracts information from one single device (e.g. Internet host, sensor node) is able to access all the secure information flows. In contrast, resilience is high if an attacker can only impersonate the device that was attacked. Unsurprisingly, a high resilience is desirable for sensor node devices.
- **Scalability.** This property is tightly linked with the amount of information that must be stored within a device in order to negotiate a secure channel with as much entities as possible. Considering that the potential Internet servers that can be accessed by a certain client (and viceversa) are huge, a key management system is not scalable if the amount of preloaded information that must be stored inside a device grows linearly with the number of potential client/servers. On the other hand, a key management system is scalable if the information required to contact any potential peer does not impose an overhead to the device. Again, it is desirable to have a high scalability.
- **Extensibility.** A key management system is not extensible if the number of peers that can be securely contacted through a negotiation process is limited to a certain number. While most key management systems are extensible (and in fact extensibility is an essential prop-

erty for most applications), there are some mechanisms that do not provide extensibility. This is not advisable for applications that have a growing number of clients and/or servers due to maintenance issues, such as a SCADA system. As a result, this property must be considered in this analysis.

3.2.3 Analysis of PKC and Pre-shared Keys

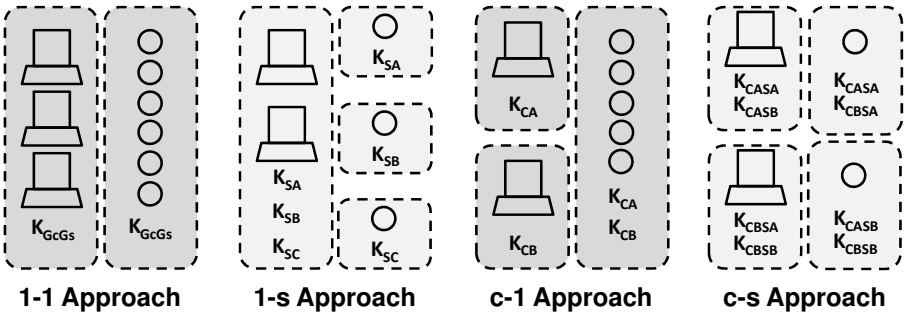


Fig. 3: Approaches for sharing secret information

In this section we will analyse the suitability of PKC and the pre-shared keys strategy to negotiate session keys between a sensor node and any external entities in an Internet context. Observe that it is possible to study different approaches for the pre-shared keys strategy. There is a very simple approach where all the members of the network share a common key (*1-1*). In addition, we can consider that all servers (*1-s*), all clients (*c-1*), or everyone (*c-s*) can have their own unique keys. All these approaches (shown in Fig. 3) will be further explained and examined in this section.

Public Key Cryptography The development and implementation of public key primitives for sensor nodes by the academic community, such as those based on Elliptic Curve Cryptography (ECC), has been a remarkable feat. The time to execute the main cryptographic operation of ECC, the scalar point multiplication, has been reduced from 34 seconds [**?**] in 2004 to less than 0.5 seconds [**?**] in 2009. With ECC, any node can make use of digital signature schemes (ECDSA), key exchange protocols (ECDH), and public key encryption schemes (ECIES) [**?**]. Note that PKC is too expensive to be used by sensor nodes implementing web servers, as its software

implementation (420 ms) is too slow for serving external requests. Also, the use of other PKC primitives with extremely efficient encryption and verification operations such as Rabin [**?**] is discouraged. For example, in TLS server nodes need to decrypt a message encrypted by the client, but decryption in Rabin is basically as costly as a decryption operation in RSA.

In spite of this, the PKC strategy provides a very good set of properties. The distribution of the credentials can be done both offline and online, as the peer (public key) certificates can be preloaded in the devices or transmitted upon request. Also, both clients and servers can be authenticated when using an adequate ciphersuite. The resilience is also high, because a subverted device usually stores only its public key and private key, thus an attacker can only impersonate that particular device. The scalability is not great but it is still good: while it is necessary to preload the certificate of a CA in order to verify all peer certificates signed by such CA, those peer certificates can be exchanged during the negotiation process. Finally, as there is no limitations on the number of peer certificates that can be validated, the extensibility is excellent.

It seems that an Internet-connected sensor node that behaves as a server should not make use of PKC. However, the situation changes if the sensor node behaves only as a client, accessing external web servers. Any external system cannot perform a DoS attack to the node by forcing it to calculate expensive PKC operations, as the node itself will make use of these operations only when it wants to open a secure connection with an external server. In terms of storage, a sensor node behaving as a client only needs the certificate of the CA that signed the certificate of the servers it accesses, and such certificate can even be retrieved online. Moreover, if the client node must authenticate himself, the only extra storage requirements are its public key and private key.

Table 2: Properties of PKC and PSK strategies

	PKC	**Pre-shared key**		
		1-1	*1-s*	*c-1*
Distribution	Online, Offline	Offline	Offline	Offline
Authentication	Client/Server	Group/Group	Group/Server	Client/Group
Overhead	ECC operations	Negligible	Negligible	Negligible
Resilience	High	Low/Low	Low/High	High/Low
Scalability	Good	High	High^{Server}	High^{Client}
Extensibility	Yes	Yes	Yes	Yes

Pre-shared keys In the pre-shared key strategy, the clients and the servers share some pre-established keying material that can be used to derive a shared key between peers. Actually, in a WSN scenario, there are multiple approaches to implement this particular strategy. For example, for external connections, a group of sensor nodes can have the same pre-shared secret (i.e. one key per WSN), or every sensor node can have its own pre-shared secret (i.e. n keys for a WSN of size n, one per sensor). The description of the different approaches is listed below, and Table 2 provides a summary of the properties of these approaches, alongside with the properties of the PKC approach.

- *1-1*. In this approach, one group G_c of clients share the same pre-shared key ($k_{G_cG_s}$) with a group G_s of servers.
- *1-s*. In this approach, every server s has its own pre-shared key (K_s). Consequently, clients must store one key per server they want to connect to.
- *c-1*. In this approach, every client c has its own pre-shared key (K_c). As a result, the servers must know in advance which are the clients that will try to connect them, and store a key per client.
- *c-s*. In this approach, every individual client c and server s share a common pre-shared key (K_{cs}). Note that this particular approach will not be analysed in the pre-shared key strategy, as it combines the disadvantages of both *1-s* and *c-1*.

One of the major drawbacks of these approaches is related to the distribution property. All secret information must be preloaded prior to any exchange of information, thus it is only possible to securely connect clients and servers that already know of each other. Another drawback is the overall resilience of the approaches. As one element (clients in *1-s*, servers in *c-1*, everyone in *1-1*) will store all the pre-shared keys, such element is the weakest link of the security chain, and any adversary that gains control of that element will take control of the whole network. Besides, it is necessary to consider that the pre-shared key will be reused every time. Not only is advisable to use a mechanism that will derive a session key from the pre-shared key, but also such key should be updated over time.

Nevertheless, the benefits of using any of the pre-shared key approaches in a WSN context are still significant. The computational overhead is negligible, as the pre-shared key is already preloaded and there is no need to engage in costly negotiations. The scalability of all approaches is mostly high, although there are some cases (the *1-s* approach for client

nodes and the *c-1* approach for server nodes) that are not advisable due to the limited memory available to the nodes. As for the authentication property, it is possible to provide either client authentication (*c-1*) or server authentication (*1-s*), although all approaches provide group authentication (e.g. if I have the pre-shared key then you know I can be trusted). Finally, the extensibility of all approaches is high, although there are some maintainability problems: in the *c-1* approach every server must be preloaded with the pre-shared key of any potential clients, and in the *1-s* approach whenever a server changes its key all its clients must also change it.

3.2.4 Analysis Results

As we have seen, PKC it is a viable option for applications without hard real-time requirements, where a slight delay (e.g. around 2-3 seconds) in the reception of the information is not critical. Observe that there are other factors that must be taken into account when considering this solution, such as the energy consumption of the node, the code size of the PKC algorithms and their memory requirements, and the expected lifetime of the application with respect to the number of connections from the node to the external server. As for pre-shared keys, they can be very useful for systems where the timing of the alarms is crucial, but they do not offer the same benefits as PKC (e.g. the key needs to be updated more frequently, thus the maintenance efforts are higher). Besides, note that the *1-s* approach is useful only if the number of external servers that must be contacted for a particular application is limited, due to memory constraints.

4 Conclusions and contributions

4.1 Conclusions

This report summarizes the advances carried out within the ARES project to study the development of authorization mechanisms in critical infrastructures.

We have studied how heterogeneous SCADA systems, which make use of different technologies such as WSNs, MANETs, and the Internet, are specially useful for the supervision of critical infrastructures. After identifying the need of an authorization gateway, we have developed a functional architecture consisting of three specific components that can manage the credentials from all the actors involved in this scenario: from

SCADA operators to substation (on-site) operators, including special sensor nodes (backbone nodes) that are proactive and can send alarms to external entities. All these components are described within this report.

4.2 List of Contributions

The contributions by members of ARES specifically related to task WP1.T4 that have been published or that are currently accepted to be published are listed below. Note that this list does not include those articles related to task WP1.T2 and WP1.T3.

A Appendix: Summary of Results of WP1.T2

This appendix summarizes the results obtained from papers developed under **WP1.T2, "Design of authentication, authorization and logging services specifically for the CII"** that serve as the foundation for the development of the *substation component* of the authorization gateway (cf. section 3.1.2). In particular, this appendix will first examine the different integration strategies (section A.1) and requirements (section A.2) that are available to connect a sensor node with the Internet, and vice versa. Section A.3 will introduce the analysis of the different integration strategies for remote substations, and section A.4 will conclude that both sensor nodes and RTUs should provide their services through the Base Station(s).

A.1 Integration Strategies

It is possible to classify the integration approaches between the Internet and WSNs in two different ways: stack-based [**?**] and topology-based [**?**]. In stack-based classification, the level of integration between the Internet and a WSN depends on the similarities between their network stacks. A WSN can be completely independent from the Internet (*Front-End*), be able to exchange information with Internet hosts (*Gateway*), or share a compatible network-layer protocol (*TCP/IP*). On the other hand, in topology-based classification the level of integration depends on the actual location of the nodes that provides access to the Internet. These nodes can be a few dual sensor nodes (e.g. base stations) located on the root of the WSN (*Hybrid*), or a full-fledged backbone of devices that allow sensing nodes to access the Internet in one hop (*Access Point*). For the sake of clarity, the different approaches (which are shown in Figure 4) will be explained in the following paragraphs.

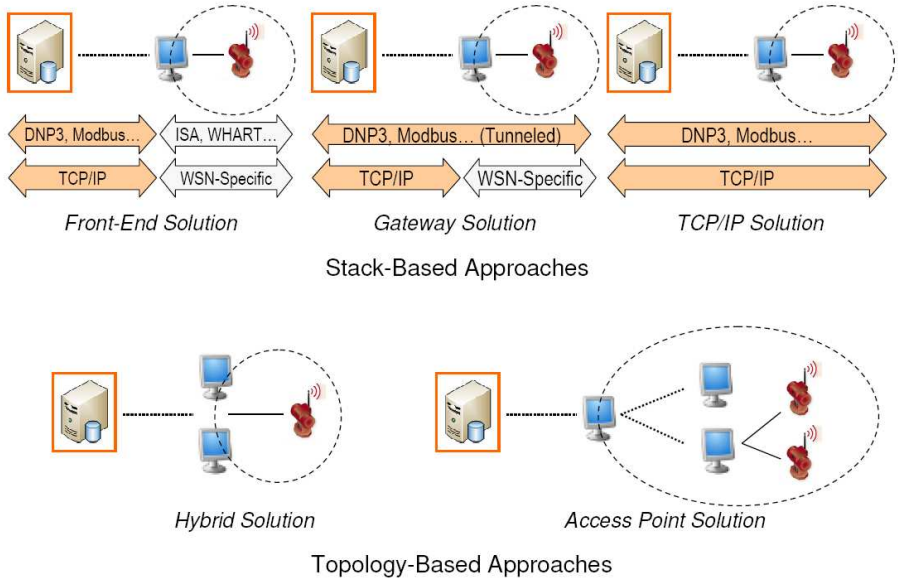


Fig. 4: Integration Strategies

In **stack-based classification**, the first approach is the *Front-End solution*. In this solution, the external control systems (e.g. the central SCADA system) and the WSNs of the substations never communicate directly with each other. In fact, the sensor network is completely independent from the Internet, so it can implement its own set of protocols (e.g. ZigBee, ISA 100.11a, or WirelessHART). All interactions between the outside world and the sensor network will be managed by an concentrator device (e.g. a RTU). This type of device is able to store all the data streams coming from the WSN, and it can also provide control systems with field information through well-known interfaces (e.g. DNP3 or Web Services). In addition, any queries coming from the control systems will always traverse the concentrator device. Most academic and commercial control systems that use the Internet use this type of solution.

The second approach, the *Gateway solution*, considers the existence of a device (e.g. a RTU) that acts as an application layer gateway, in charge of translating the lower layer protocols from both networks (e.g. TCP/IP and proprietary) and routing the information from one point to another. As a result, Internet hosts and sensor nodes can be able to exchange information without establishing a direct connection. For example, nodes

will be able to answer specific protocol queries (e.g. DNP3, HART) from external control systems. In this solution, the sensor network is still independent from the Internet, and all queries still need to traverse a gateway device. As of 2010, this solution is technically possible with standards like ISA100.11a that support protocol tunneling (e.g. using a "tunnel" object).

As for the third approach, the *TCP/IP solution*, sensor nodes implement the TCP/IP stack (or a compatible set of protocols such as 6LoWPAN [**?**] in 802.15.4 networks), thus they can be considered as full-fledged elements of the Internet. Any Internet host (e.g. the elements of a control system) can open a direct connection with them, and vice versa. In fact, this solution fully integrates industrial WSNs with the IoT. However, using this approach, it is not possible to use specific substation protocols like WirelessHART in the WSN, as these protocols define their own stacks. Still, we can use other protocols that support TCP networks, like DNP/IP or Modbus/TCP.

Regarding the **topology-based classification**, the *Hybrid solution* approach considers that there are a set of nodes within the WSN, usually located in the edge of the network, that are able to access the Internet in a direct way. In fact, these nodes can be easily mapped to base stations, since every sensor within the WSN needs to traverse them in order to connect the central system, and vice versa. The specific features of this type of approach are redundancy and network intelligence. By default, this approach considers that it is possible to provide more than one "base station" to access the functionality of the network. Besides, as those "base stations" have the capability to connect the Internet, it means that the intelligence of the network (i.e. the implementation of the different substation protocols) is pushed onto a subset of the WSN.

This delegation of capabilities is further developed in the *Access Point solution* approach. Here, WSNs become unbalanced trees with multiple roots, where leaves are normal sensor nodes and all other elements of the tree are Internet-enabled nodes. As a result, all sensor nodes are able to access the Internet in just one hop. One of the main features of this approach is the possibility to increase the capabilities of nodes that belong to the backbone network. For example, backbone nodes can have more resources than normal nodes, and can implement faster network standards (e.g. 802.11 vs. 802.15.4).

It is important to note that the previously shown topology-based networks are usually combined with the approaches from the stack-based classification. For example, in a backbone-type network, the Internet-

enabled nodes can behave i) as a front-end, effectively isolating the WSN sensors from the Internet, or ii) as gateways, allowing direct data exchange between sensors and the central system. There is an exception, though: it is essentially irrelevant to combine the TCP/IP solution with the hybrid and backbone solutions, as every node is able to connect the Internet Internet. In fact, the only task of the nodes that connect the Internet with the local network will be to behave as translators (e.g. between 6LoWPAN and IPv6).

A.2 Requirements

For studying the security of the interconnection between industrial WSNs and the Internet, it is essential to consider not only security requirements, but also the requirements that such control networks must satisfy, like maintenance, system performance and reliability of the resources/services. The reason is simple: Some of these requirements have a direct influence on the security requirements of the network, and vice versa. For example, if we use an end-to-end secure channel to open a secure channel between a sensor node and a central system, we will increase the overhead associated to the node, not only in terms of response time but also in terms of the memory available to the node.

Maintenance. One important aspect in the management of any substation system is the maintenance of its software and hardware elements. To prevent the appearance of errors and mistakes, every device must be properly configured at all times, and periodical tests should be run either from the control center or in the substation itself. Moreover, the software components included within the devices should be up-to-date (after such components have been properly tested in a controlled testbed), and new hardware devices should be added to the substation if needed. Consequently, the properties associated to maintenance are:

- *Addressing.* It is necessary to specify some kind of unique identification (e.g. network address) for every RTU present in the substation in order to access the stream of data each one produces. This property is related to how the different identifications of the devices are accessed and who is in charge of storing those Identities (IDs).
- *Internal Access.* The services offered by the devices found inside the substation should be accessed locally by substation operators, either for testing purposes or for redundancy purposes. This property is concerned with the actual complexity of accessing the devices of the substation locally.

- *Maintainability.* As with any device, the software included within the RTUs will need to be upgraded for many reasons, such as upgrades, optimizations, security patches, and so on. This property refers to the number of devices that must be changed in order to fix or upgrade the functionality of the substation.
- *Extensibility.* The number of RTUs that can be found in a certain substation will surely change during the lifetime of the infrastructure. As a property, extensibility is related to the overall changes that must be done in the substation in order to include new hardware devices.

Reliability. As one of the major purposes of a substation is to examine and control the state of critical infrastructures, the functionality provided by the substation must be reliable enough to offer its services within certain quality levels. The data streams provided by the RTUs should be available at all times, and any query regarding the actual contents of a certain data stream should arrive at the central system as fast as possible in order to react to critical situations. Consequently, the properties associated with reliability are:

- *Availability*[3]. As the infrastructures monitored by the substations are usually critical, the data produced by the RTUs must be available at all times in order to react to problematic situations and ensure the integrity of the whole system. As a property, there are in fact two dimensions of availability: one related to reliability (using the redundancy of the system to avoid single points of failure) and one related to security (existence of denial of service attacks and use of self-healing mechanisms to provide the services even in the case of attacks / system failures).
- *Performance.* Not only must the data be available at all times, but it must also be retrieved from the RTUs at an acceptable speed. As a property, performance is related to the hardware capabilities of the devices of the substation, in addition to the actual speed of the substation network infrastructure, and the number of hops between the RTU and the data repository.

Overhead. It is necessary to achieve a balance between the number of resources available to a device and its overall cost. A device should not be

[3] Note that availability can be considered as a security requirement, but it has been classified as a reliability requirement due to its close relationship with the functional dimension of a substation.

encumbered by an excess of workload, but it should not have any unnecessary resources. Besides, those resources should be optimized to work in the substation environment. Consequently, the properties associated with the overhead are:

- *Device Resources.* In order to implement the different protocols that provide the core functionality of substations, such as DNP3 or WirelessHART, the devices must use some of their HW and SW resources. This property refers to the amount of resources that are needed within a node to implement those protocols.
- *Optimization.* There are some specific protocols that are optimized to provide the best possible functionality in a particular environment. This property is related to the existence of network-specific protocols (such as WirelessHART), which are aware of the specific features of the network environment and use them to provide better services (e.g. network redundancy and link robustness).

Security. Ensuring the security in the different processes of a substation is a matter of utmost importance. If security is not fully considered, any problem that affects the integrity of the elements of a substation will potentially have an influence on the real world as well, harming not only physical infrastructures but also human lives. Therefore, only authorized users must have the right to modify the state of the elements of a substation, and only trusted users must be able to access the streams of data produced by the substations. In addition, there should exist some mechanisms that store the interactions between the different elements of the substations, to facilitate not only the analysis of the behaviour of the system but also the detection of possible security breaches. Consequently, the properties associated with security are:

- *Secure channel.* Whenever two devices that belong to the same SCADA system (e.g. a machine from the central system and an RTU from a substation) communicate, it is important to set up a secure channel that supports end-to-end integrity and confidentiality services. If the integrity of the data stream is protected, attacks will not be able to falsify any readings. In addition, once the confidentiality of the information flow is assured, adversaries will be unable to read any sensitive information. As a property, it refers to the type of machines and mechanisms that are involved in the creation of a communication channel that support confidentiality and integrity.

- *Authentication.* As for user authentication, the devices should be confident about the identity of the user that is requesting a certain operation. As a property, authentication is also concerned with the location and the nature of the mechanisms and elements that can be used to prove the identity of a human user.
- *Authorization.* Once any user of the network (be it either a human user or a machine) proves their identity, it may be necessary to check whether that user has the rights to access the information. Not only the access to the data should be controlled, but also the granularity of the data, as well. Beyond data, it is also necessary to monitor control operations (e.g. devices must only be reprogrammed by authorized users). As a property, authorization deals with the types of mechanisms, credentials and tools that can be used to check whether a certain entity is authorized to perform an operation.
- *Accountability and Detection.* Since a heterogeneous set of users will be accessing the services of a substation, it is important to record the interactions with those users. By storing all interactions, we can recreate security incidents and abnormal situations. In addition, we can detect specific attacks in real time. As a property, accountability and detection refers to the structure of the accountability subsystems and the mechanisms that can be used to analyze them.

A.3 Analysis

Once we have introduced the integration strategies and the requirements of industrial WSNs, we should be able to tackle these two questions: i) "What are the specific advantages and disadvantages of every integration strategy in the context of industrial WSNs?", ii) "Which strategy should I choose for a particular deployment?". These questions will be answered by discussing the influence of the integration strategies over the previously presented requirements. This is summarized in Table 3 and Table 4 for the sake of clarity.

Maintenance. The properties associated with maintenance that have to be analyzed are addressing, internal access, maintainability, and extensibility.

In terms of *addressing*, the 'Front-End' and 'Gateway' solutions require translating the identity of the node (e.g. Metering Pump A, DNP3 Address 65519) to the actual address of the node (e.g. WirelessHART EUI-64 Address). The translation table should be located within the remote substation, as the conversion between identity and WSN address will

Remote Substations		*TCP/IP*
Maintenance	*Addressing*	Translation at central system
	Internal Access	Use IP address
	Maintainability	Update all nodes
	Extensibility	Add row to translation table
Reliability	*Availability*	Vulnerable due to constraints
	Performance	Extra penalty if packet processing
Overhead	*Device Resources*	More mechanisms inside nodes
	Optimization	Use only IP and MAC layer services
Security	*Secure channel*	End-to-end
	Authentication	Distributed mechanism
	Authorization	Distributed mechanism
	Accountability	Limited by storage
	Detection	Efficient detection rules

Table 3: Analysis of Properties and Integration Strategies (I)

Remote Substations		*Front-End*	*Gateway*
Maintenance	*Addressing*	Translation at substation	
	Internal Access	Use local services	
	Maintainability	Update 1+ device	Update all nodes
	Extensibility	Add row to translation table and local management	
Reliability	*Availability*	Single point of failure, “store and forward”, “cache”	
	Performance	May need of ‘Access Point’ solution. Extra penalty if packet processing	
Overhead	*Device Resources*	Less mechanisms	More mechanisms inside nodes
	Optimization	Take advantage of WSN-specific optimizations	
Security	*Secure channel*	Bridged at base station	End-to-end
	Authentication	Centralized mechanism	Distributed mechanism
	Authorization	Centralized mechanism	Distributed mechanism
	Accountability	Centralized	Centralized, only statistics
	Detection	Lightweight detection rules	Efficient detection rules

Table 4: Analysis of Properties and Integration Strategies (II)

be performed there. On the other hand, the ‘TCP/IP’ solution requires the translation table to be located in the central system (e.g. Metering Pump A $\rightarrow$ a.b.c.d), as such a system must use the IP addresses of the WSN nodes to open a direct connection. For this particular property, the suitability of the approaches depends on the kind of management prefered (decentralized or centralized). Note that the complexity of the addressing management increases if we take into account the ‘Hybrid’ and ‘Access Point’ solutions, as we need either to replicate the translation tables among the Internet-enabled nodes or to create a centralized service that provides a translation interface.

Internal Access is not an issue for most solutions. In all solutions, the human operators performing maintenance processes within the remote substation can use the substation network to connect with the data retrieval services (e.g. through a RTU, using TCP/IP direct connection

with the sensor nodes, etc). If the operators are in the field where the sensor nodes are deployed, they can also use the local services of the WSN-specific protocols. For example, in solutions where the WSN is independent from the Internet ('Front-End' and 'Gateway'), an operator can use the features offered by internal protocols like WirelessHART to access the data stream of a sensor node in a direct manner. As for the 'TCP/IP' solution, direct local access is also possible, although operators should know the IP addresses of the nodes they want to access beforehand. The 'Access Point' solution may add a small amount of complexity to this process, as operators need to be physically near the node they want to read data from if they want to use the internal protocols of the WSN.

Maintainability is directly related to the number of devices that need to be upgraded when a SW update is tested and accepted by the central management. In every solution, upgrading the protocol used in the WSN (e.g. ISA100.11a, TCP/IP) means upgrading all sensor nodes. Therefore, we will focus on the upgrades that target the control protocols that interface with the central system (e.g. DNP3). For control protocols, the 'Front-End' solution is the most simple to maintain: there is only one device (the concentrator device) that needs to be upgraded. On the 'Gateway' and 'TCP/IP' solutions, we need to change the control protocols in all sensor nodes. Nevertheless, in the 'Front-End' solution the whole WSN will not be available during the upgrade (i.e. the concentrator device is the only entry point to the network), while for the other two solutions it is possible to perform a gradual upgrade. Note that the 'Hybrid' solution is also able to provide support for gradual upgrades due to its inherent redundancy, while the 'Access Point' solution can not provide full support for gradual upgrades as every sensor node is usually connected to one single backbone node.

Finally, regarding the *Extensibility* property, adding a new node is not a very cumbersome task. In the 'Front-End' and 'Gateway' solutions, we need to include a new entry in the translation table and run the specific mechanisms of the WSN protocols. The process in the TCP/IP solution is simpler, as the only change that needs to be made is to add a new entry to the translation table. The task is similar for the 'Hybrid' and 'Access Point' solutions, although, if the translation table is distributed then all changes must be stored in all devices (or in a centralized service if a translation interface is available).

Reliability. The properties associated with reliability that have to be analyzed are availability and performance.

In terms of the *Availability* property, the 'Front-End' solution is weak against failures or attacks (e.g. DoS attacks). As there is only one single point of entrance to the WSN, any problem will bring the whole system down. Still, this solution can be improved if combined with the 'Hybrid' solution, as redundancy improves availability. Besides, the 'Front-End' solution can make use of the lack of integration with the WSN to transparently implement self-healing mechanisms. For example, the concentrator device can use "store and forward" mechanisms, and can also know whether a certain sensor of the WSN is unreachable and try to obtain information from another sensor if the WSN is redundant enough. The 'Gateway' solution has the same advantages and disadvantages of the 'Front-End' solution, although the self-healing mechanisms will be less transparent since data messages will arrive "as is" to the sensor nodes. It also must take into account attacks that target the application layer. Finally, the 'TCP/IP' solution is very vulnerable against attacks that target the availability of the network, mainly due to the limited capabilities of the sensor nodes (i.e. an attacker will need less resources to DDoS a node with just 128KB of memory), so it will be indispensable to implement protection mechanisms in the remote substation access points. This particular problem is shared by the 'Access Point' approach, as the backbone will use TCP/IP to transmit information to the sensor nodes.

In contrast, the 'Access Point' approach is has some advantages in terms of *Performance*. If the backbone nodes use high-speed communication technologies (and have a reliable power supply), the data streams can be provided to the substation network at a very fast speed. To put this assertion into context, the maximum data rate of the 802.15.4-based WirelessHART and ISA100.11a protocols is 250 Kbit/s, while the maximum data rate of 802.11b-based networks is 11 Mbit/s. Of course, as the link between the backbones node and the sensor nodes has a low data rate, all the other solutions can have a similar performance if there is only one hop between the sensor node and the substation network. Performance can be also improved if the central system does not want to access real-time data, as the 'Front-End' and 'Gateway' solutions can prefetch data streams and store them in a cache. Observe that packet processing may also harm the overall performance of the WSN, thus all solutions that impose any extra packet processing (e.g. 'Front-End' solution) may have a performance penalty.

Overhead. The properties associated with overhead that have to be analyzed are device resources and optimization.

In terms of *Device Resources*, all solutions that push the intelligence to the sensor nodes (e.g. the 'Hybrid' and 'Access Point' solutions, the 'TCP/IP' solution) require that sensor nodes have enough capabilities to implement the application protocols, including any security protocols. It should be pointed out that sensor nodes that are used in SCADA systems are only slightly better that the sensors used in the academic world (96 KB RAM, 128 KB serial flash, 26 MHz microcontrollers, 80 KB ROM). Therefore, most results regarding the feasibility of implementing a complete IP-stack in sensor nodes as well as issues related to computational and memory constraints available in literature can be extrapolated to current industrial nodes. Analyzing the capabilities of industrial sensor nodes, it would seem difficult to implement a TCP/IP or WSN-specific stack, a control protocol parser, and all the necessary security mechanisms (cryptography primitives, link-layer security, end-to-end security) inside the same node.

As for the *Optimization* property, all solutions that make extensive use of WSN-specific protocols ('Front-End', 'Gateway', 'Hybrid') can benefit from their optimizations. For example, WirelessHART uses a TDMA data-link layer to provide Quality of Service, supports mechanisms like channel hopping to maximize coexistence with other ISM band equipment, and also implements a self healing, redundant path mesh routing protocol. Some of these benefits cannot be found in pure TCP/IP networks, and other benefits (e.g. the use of an underlying data-link layer that provides certain properties) are usually not explicitly considered. Nevertheless, the 'Access Point' solution can also obtain some benefit from these optimizations, since the connection between the sensor nodes and the backbone nodes uses the WSN-specific protocols.

Security. The properties associated with security that have to be analyzed are secure channel, authentication, authorization, and accountability / detection.

In order to comply with the *Secure Channel* property, it is necessary to protect the confidentiality and integrity of all communications between the central system and the sensor nodes. The 'TCP/IP' solution is able to provide an end-to-end secure channel between these entities, as every device located in the routing path will use the TCP/IP stack. Still, it is usually not possible to use IPsec due to constraints on the WSN nodes [**?**], so it is necessary to use other mechanisms such as SSL/TLS

at the transport layer or WS-SecureConversation at the application layer if web services are used. These security mechanisms at the application layer can also be used by the 'Gateway' solution due to its forwarding capabilities. The 'Front-End' solution cannot create an end-to-end secure channel, but the information exchange can be easily protected: one part of the connection will make use of TCP/IP security mechanisms and the other part of the connection will employ the WSN-specific protection mechanisms. It should be noted that, for both this 'Front-End' solution and the 'Gateway' solution, it is also possible to create a Virtual Private Network between the central system and the gateway/front-end located in the remote substation.

Regarding *Authentication*, one of the major challenges to solve in user authentication is the location of both the authentication service and the storage of the user credentials (e.g. user/password pairs). In the 'Front-End' solution, where all traffic must traverse one single device, all processes and user data can be stored in that device. The other solutions ('Gateway', 'Hybrid', 'Access Point', 'TCP/IP') have multiple points where the service can be provided. As a result, it is necessary to use protocols and mechanisms such as Kerberos [**?**] in order to centralize the authentication information and avoid replication. It should be pointed out that in battery-powered nodes these centralized approaches can become energy-consuming if the sensor nodes must use an external service to test the user credentials. Consequently, it can be also possible to replicate the user databases if needed, although this configuration increases the complexity of the maintenance processes. Another approach that can be used for the 'Gateway' solution, which also forces all traffic to traverse one single device, is to implement a mechanism where an user can obtain a dedicated secure channel between himself and the gateway after the authentication process.

Authorization is very similar to authentication. Its main challenge is the location of the authorization service and the permissions of users. The same solutions explained in authentication apply for authorization, although it should be noted that the maintenance of a distributed authorization database is more complex. User permissions change more frequently than user identities.

As for *Accountability*, one possible approach is to use a single entity to store all the interactions between the central system and the sensor nodes. This approach is quite optimal for centralized solutions such as 'Front-End' and 'Gateway', because in other solutions any interaction information must be collected from the different entities and sensor nodes.

However, if end-to-end security mechanisms are used (e.g. in the 'Gateway' solution), the gateway devices can only extract statistical data from the information flow. As for pure decentralized solutions, where the interactions are stored in all sensor nodes, it may not be advisable if there is storage limitations within the nodes. Finally, regarding *Detection*, decentralized solutions that push intelligence to the sensor nodes ('Access Point', 'TCP/IP') need to implement all the detection rules within all the sensor nodes, because any node can become a target of attacks. This can be cumbersome for the nodes if they have few computational resources, thus other centralized approaches may be advisable. Nevertheless, for all solutions, the creation of lightweight detection rules within the WSN that can detect possible malfunctions and internal attacks should be recommended.

A.4 Discussions and Conclusions

Once the features of the different integration strategies have been analyzed, it is time to discuss their suitability for industrial environments. Due to the importance of the 'TCP/IP' solution for the nascent Internet of Things (IoT) paradigm, this solution will be discussed first, followed by the 'Front-End' solution and the 'Gateway' solution. For the sake of clarity, these discussions are summarized in Table 1.

The 'TCP/IP' solution guarantees that the WSN located in remote substations are fully integrated with the Internet, but it is not clear whether this can be considered as an advantage or not. In terms of security, it is necessary to protect the WSN from any kind of intrusion, as even an increase on the network traffic can become problematic for the sensor nodes due to their limited capabilities. Other security aspects like user authentication and authorization have no established solution, but can be solved using mechanisms like Kerberos. Besides these security issues, there are other aspects in the 'TCP/IP' solution that need to be considered. In particular, a TCP/IP-based WSN will not benefit from the specific optimizations of protocols like ISA100.11a, and will have no native support for "store and forward" mechanisms and data streams caches. Besides, the capabilities of the sensor nodes may not be enough to implement the required protocols. Nevertheless, the 'TCP/IP' solution also have some specific advantages, such as support for gradual updates (i.e. updating one node will not bring the entire system down) and resilience to device failure (i.e. a failure in one node will probably not endanger the whole network).

In contrast, the 'Front-End' solution solves some of the problems of the 'TCP/IP' solution, although it also has issues of its own. Existing standards can be used to implement the security mechanisms, although the existence of a concentrator as an entry point of the network makes this solution quite vulnerable against availability attacks. This can be solved by using the 'Hybrid' and 'Access Point' solutions, but these solutions have their own specific problems that must be considered in a network deployment (mainly due to the replication of resources). Another important benefit of the 'Front-End' solution is the use of the WSN-specific optimizations, and the ability to include self-healing mechanisms (e.g. if one node is not available we can access another one if the WSN is redundant enough). Finally, the maintenance of the network is quite simple (e.g. only one device needs to be upgraded), but this is a doubled-edged sword, as the network will not be available during the upgrade process. Nevertheless, this issue can be solved through replication and the 'Hybrid' and 'Access Point' solutions.

The 'Gateway' solution provides a middle ground between the 'TCP/IP' solution and the 'Front-End' solution. It has some of the 'Front-End' solution benefits (e.g. use of WSN-specific optimizations, implementation of "store and forward" mechanisms), and it allows the central system to query the sensor nodes directly. Nevertheless, it also pushes some complexity to the sensor nodes, and it also needs to solve certain security details such as the implementation of the authentication and authorization mechanisms. Moreover, the gateway device should parse all incoming messages in order to analyze the queries and to avoid application-specific attacks, and other aspects (such as maintainability) get more complex as well. Note that this solution can also be combined with the 'Hybrid' and 'Access Point' solutions to obtain benefits such as redundancy, although the specific problems of these solutions (e.g. distribution of tables and resources) need to be taken into account.

From the previous discussions, *it can be deduced that the 'Front-End' solution combined with approaches that provide extra redundancy is good enough for the present needs of the industry.* Consequently, for the development of the *substation component* of the authorization gateway (cf. section 3.1.2), we will consider that the sensor network is isolated from external networks. Nevertheless, having a sensor node proactively connecting an external system (e.g. for raising a specific and urgent out-of-band alarm) is a desirable feature, and we will consider it in the *sensor component.*

B Appendix: Summary of Results of WP1.T3

This appendix summarizes the results obtained from papers developed under **WP1.T3, "Development of delegation services for distributed authorization"** that serve as the foundation for the development of the *collaboration component* of the authorization gateway (cf. section 3.1.2). In particular, this appendix will first explain the existing problems of controlled delegation by overviewing the state of various delegation mechanisms (section B.1). Afterwards, we will introduce the concept of Attribute Federation (section B.2), which enables authorized actors to access different infrastructures with different policies. Finally, we will describe how it can be possible to associate a quota to the delegation process in section B.3.

B.1 Controlled Delegation

Delegation is needed to obtain a real scalable distributed authorization. However, the uncontrolled use of delegation statements can become a security threat because any user could improperly get the same privileges over a resource than the owner of that resource. Therefore, delegation solutions must include a mechanism to control the delegation and to produce appropriate authorizations statements. The goal of this section has been to study and put into perspective the delegation implications of a group of schemes that have been proposed as solutions for distributed authorization problems. The results are summarized below:

- In *PolicyMaker* and *Keynote* the delegation statement does not exist, and any authorization statement can be delegated again and again without any control.
- *SDSI* considers three different possibilities to control the delegation, although *SPKI* has reduced it to a boolean condition. Such a boolean parameter is only a modest mechanism to control the depth of the delegation.
- The *PMI* solution provides more complex mechanisms to perform the delegation, allowing the possibility to use the extension fields of the attribute certificate for a controlled delegation.

B.1.1 PolicyMaker and Keynote

Blaze, Feigenbaum and Lacy introduced in [**?**] the notion of Trust Management. In that original work, they proposed the PolicyMaker scheme

as a solution for trust management purposes. PolicyMaker encodes trust in assertions. They are represented as pairs (f,s), where s is the issuer of the statement, and f is a program.

Trust is monotonic in PolicyMaker, that is, each policy statement or credential can only increase the capabilities granted by others. Moreover, trust is also transitive. This means that if Alice trusts Bob and Bob trusts Carol, then Alice trusts Carol. In other words, all authorizations are delegable, thus it is not possible to restrict delegation capabilities. This is the reason why delegation is uncontrolled in this scheme.

Keynote [**?**] is a derivation of PolicyMaker, and has been supported by IETF. It has been proposed and designed to improve aspects of PolicyMaker. Keynote uses a specific assertion language that is flexible enough to handle the security policies of different applications. Assertions delegate the authorization to perform operations to other principals. KeyNote assertions are composed of five fields, Authorizer, Licensees, Comment, Conditions and Signature.

Given a set of action attributes, an assertion graph is a directed graph with vertex corresponding to principals. An arc exists from principal A to principal B if an assertion exists where the Authorizer field corresponds to A, the Licensees field corresponds to B and the predicate encoded in the Conditions field holds for the given set of action attributes. A principal is authorized, within a given set of action attributes, if the associated graph contains a path from a policy to the principal. We conclude then that all authorized principals are allowed to re-delegate their authorizations. Thus, there is no restriction on delegation.

B.1.2 SDSI/SPKI

This solution is a unification of two similar proposals, SDSI (Simple Distributed Security Infrastructure) and SPKI (Simple Public Key Infrastructure). SPKI was proposed by the IETF working group and, in particular, by Carl Ellison [**?**]. SDSI, designed by Ronald L. Rivest and Butler Lampson [**?**], was proposed as an alternative to X.509 public-key infrastructure. The SPKI/SDSI certificate format is the result of the SPKI Working Group of the IETF [**?**].

SDSI establishes four types of certificates: Name/Value, Membership, Autocert and Delegation. We focus on the description of Delegation Certificates, because these are the mechanisms for implementing the delegation in SDSI. SDSI provides two types of delegation, based on the structure of the delegation certificate (figure 5):

– The issuer can delegate to someone by adding that person as a member of the group which he control. 'A' issues a delegation certificate to 'B'. Therefore, 'B' will have the same privileges as group.
– or the issuer can delegate to someone so that this person is able to sign objects of a certain type on the user's behalf. The "certain type" is defined by using the template form.

```
(Delegation-Cert:
(Template: form )
(Group: group1 )
(Signed: ...))
```

Fig. 5: Delegation certificates

SPKI/SDSI unifies all types of SDSI certificates into one single type of structure (figure 6).

```
(cert
(issuer <principal>)
(subject <principal>)
(propagate)
(tag <tag>)
(valid ))
```

Fig. 6: Authorization Certificates

The field propagate is used to perform the delegation. As it was desirable to limit the depth of delegation, initially, SPKI/SDSI had three options for controlling this: no control, boolean control and integer control. Currently, these options have been reduced to boolean control only. In this way, if this field is true, the Subject is permitted by the Issuer to further propagate the authorization.

B.1.3 Privilege Management Infrastructure (PMI)

Similarly to PKI [**?**], ITU-T has defined the attribute certificates framework for authorization services. It defines the foundation upon which a Privilege Management Infrastructure (PMI) can be built [**?**].

One of the advantages of an attribute certificate is that it can be used for various purposes. It may contain group membership, role, clearance,

or any other form of authorization. In fact, ITU-T defines PMI three models for different environments, Control, Roles and Delegation model.

The delegation model has four components: the privilege verifier, the Source of Authorization (SOA), the Attribute Authorities (AAs), and the privilege asserter. The SOA, the initial issuer of certificates, is the authority for a given set of privileges for the resource and can impose constraints on how delegation can be done. The SOA assigns privileges to intermediary AAs, which further delegate privileges to other entities but obviously not more privilege than they hold. A delegator may also further restrict the ability of downstream AAs. If the privilege asserter's certificate is not issued by that SOA, then the verifier shall locate a delegation path of certificates from that of the privilege asserter to one issued by the SOA. The validation of that delegation path includes checking that each AA had sufficient privileges and was duly authorized to delegate those privileges. Using the extension mechanism of PMI is possible to design of a controlled and scalable delegation model [**?**].

B.2 Attribute Federation

While the extension mechanism of PMI allow us to create a delegation model, the attribute certificates used in this model can be considered "global": well-known attributes known by an entire infrastructure. However, in a Critical Infrastructure there might exist local policies that define local attributes. The combination of these global and local attributes, and the delegation of privileges between infrastructures (e.g. an external auditor needs to check some data) is an open challenge that has been solved by implementing the concept of *Attribute Federation.*

The concept of Attribute Federation is similar to Identity Federation but applied to Authorization sentences, avoiding identity information exchange. In each Federation, a new element has been defined named Attributed Ontology Server (AOS), in which entities store their attributes and the relationships between them, i.e. attribute subscriptions, to simplify the authorization process. The AOS is independent from the underlying PMI and can be easily distributed. It allows linking attributes from different users and reuse them in the definition of authorization policies.

B.2.1 Underlying Technologies

Before explaining the concept of Attribute Federation, we need to introduce the different technologies that form the bases of this concept.

OWL. The Web Ontology Language (OWL) has been derived from the DAML+OIL Web Ontology Language [**?**] and is build on earlier W3C standards such as the Resource Description Framework (RDF) language, that is used to represent information about resources in the World Wide Web and then it has, like RDF, a XML syntax. In essence, OWL adds more vocabulary for describing properties and classes.

OWL provides three increasingly expressive sublanguages:

- OWL Lite is mainly intended for classification hierarchy and simple constraints. It supports cardinality constraints, but only permits cardinality values of 0 or 1. Owl Lite is the sublanguage with the lower formal complexity from the three.
- OWL DL provides more expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). The DL from the acronym comes from its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL.
- OWL Full provides maximum expressiveness but without computational guarantees in contrast with OWL DL.

The main concepts of an OWL ontology are: Class, Invidual and Property. Classes are related with other classes by the subclass relation, individuals are instances of classes and properties are used to state relationships between individuals or from individuals to data values.

RT Framework. Li et al. proposed logic programming as a way to model authorization and delegation relations [**?**]. Although they use *Roles* for this purpose, their roles can also be interpreted as attributes, as it is commented in their work. They define a full general framework, RT for *Role Based Trust Management*. It comprised of five different solutions, each of them with different characteristics. Roles can be interpreted as privileges or attributes. The RT Framework defines a partial order in roles, establishing how rights can be inherited.

RT defines several types of credentials, the basic ones are:

1. $A.R \leftarrow D$: *D has the attribute $A.R$.*
2. $A.R \leftarrow B.R_1$: *if B says that an entity has the attribute R_1, then A says that it has the attribute R.*
3. $A.R \leftarrow A.R_1.R_2$: *if A says that an entity B has the attribute R_1, and B says that an entity D has the attribute R_2, then A says that D has the attribute R.*

4. $A.R \leftarrow B_1.R_1 \cap B_2.R_2 \cap \cdots \cap B_n.R_n$: *A believes that anyone who has all the attributes* $B_1.R_1$, ..., $B_k.R_k$ *also has the attribute* R.

RT defines an attribute federation using linked roles. In this way, users can link their attributes to other users' attributes, defining then their authorization policies in terms of other users attributes. However, it is not clear how and where credentials and authorization policies are stored or who defines them.

SAML. The Secure Assertion Markup Language (SAML), developed by the Security Services Technical Committee of OASIS, is an XML-based framework for communicating user authentication, entitlement, and attribute information. It is used in many security applications.

SAML is defined using XML [**?**] sentences whereas X.509 attribute certificates (cf. Section B.1.3) are defined using ASN.1 [**?**]. A performance comparison between both technologies was presented in [**?**], showing that ASN.1 encode, decode and file space is more effective than XML and, therefore, SAML. On the other hand, XML is a text-based format that provides a standard notation for serialized data. This fact makes XML has numerous advantages over ASN.1 which is a binary format. For this reason, we establish a mixed system using SAML and attribute certificates technologies.

The structure of X.509 attribute certificate and a brief example of its corresponding ASN1 encoding is shown in figure 7 (for detailed information see [**?**]).

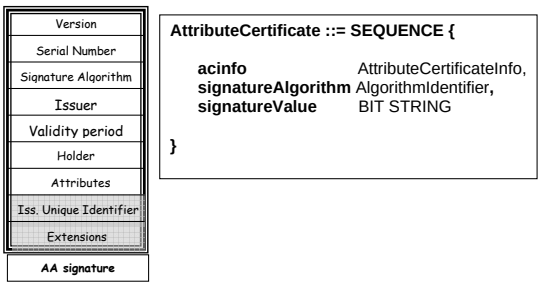


Fig. 7: X509 Attribute Certificate and ASN1 representation

Additionally, an example of SAML sentences is shown in figure 8. Furthermore, the relationship between SAML and X509 attribute certificate fields is described in the table 5.

```
<saml:Assertion ...>
 <saml:Conditions .../>
 <saml:AttributeStatement>
  <saml:Subject>
   <saml:NameIdentifier
     SecurityDomain=\www.uma.es"
     Name=\Subject1" />
  </saml:Subject>
  <saml:Attribute
    AttributeName=\Pass_Test1_Subject1"
    AttributeNamespace=\http://www.uma.es">
    <saml:AttributeValue>
       Pass_Test1_Subject
    </saml:AttributeValue>
  </saml:Attribute>
 </saml:AttributeStatement>
</saml:Assertion>
```

Fig. 8: SAML example

SAML Assertion	Attribute Certificate
Issuer	Issuer of Attribute
Subject of the Query	Certificate Subject
Validity period	AC Validity period
Attributes	AC Attributes

Table 5: Relationship between SAML and AC fields

XSAML. XSAML is a novel technology that allows us to translate the information stored in attribute certificates to SAML code and vice versa. XSAML is able of bridging the gap between the Global Attributes and Domain (Local) Attributes. The Global attributes are defined using X.509 attribute certificates whereas the Domain attributes are defined using SAML. Therefore, XSAML symbolize the glue in our infrastructure because makes possible the translation process between Global and Domain Attributes.

Figure 9 shows a brief description of XSAML system. The main design goal has been to facilitate suitability to any environment. In this way, when a task depends on the environment, an interface will be created and programmers will be able to add code according to the requirements of the environment in question.

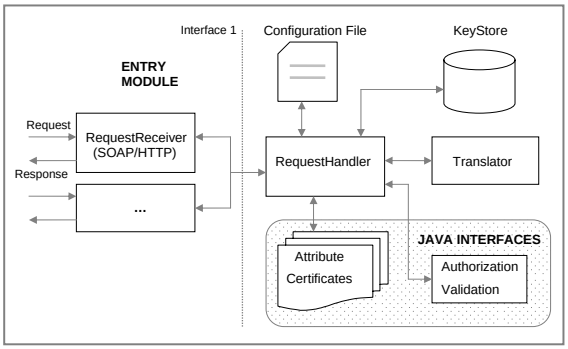


Fig. 9: XSAML components

The main components of the proposal are:

- **Entry Module** holds a servlet, named RequestReceiver. This class is implemented to exchange messages with the Service Provider (client) using SOAP over HTTP. Programmers can implement other classes to achieve these tasks. When the RequestHandler processes the request, it sends a response to the Entry Module (in this case RequestReceiver or another class designed).
- **RequestHandler** is the main class for processing the request. It allows programmers to create new classes in the Entry Module; in other words, new ways to exchange messages with the Service Provider.
- Finally, we describe the **java interfaces**. They make the system configurable. These interfaces are used to authorize the requester (interface Authorization), validate requester's certificates (interface validation), and access the attribute certificates (interface AttrCertHandler).

B.2.2 Attribute Federation: Overview

In the following paragraphs, we finally provide an overview of the concept of Attribute Federation (AF). AF is required when several ubiquitous

service providers share a common context. The concept introduced is explained using the following scenario.

Let us suppose a Power plant where different operators have different roles. For example, switchboard operators (operators) control the central switchboard to distribute power output among generators and to regulate the flow of power to outgoing power lines, and switchboard operator assistants (assistants) compile gage readings and perform other tasks as directed by the Switchboard Operator in electric generating plants. Their devices may act as service providers, e.g. providing information about a particular generator. In this case, the Power Plant defines some generic attributes with or without parameters, e.g. Assistant, Assigned(Generator), Operator, Managing (Generator) and so on, that help characterizing entities at the Power Plant.

These types of attributes are named as *Global Attributes* and are defined and issued by Attribute Authorities. The *Attribute Authority* (AA) could be part of an existing infrastructure such as for instance an X.509 (PMI) [**?**]. The definition of this type of Attribute must be reduced because in ubiquitous scenarios the global infrastructure should only be used in specific situations. The power plant is in charge of defining and issuing those attributes to Operators and Assistants, therefore each power plant acts as an AA inside the PMI.

Once those basic attributes have been issued, some Operators may need to define new attributes such as, $GenA_Junior$ or $GenA_Senior$. These specific attributes can be used to control whether a certain assistant has experience managing a class of generator. These attributes are named as *Domain Attributes*. The Domain Attributes are defined locally and its meaning is limited to the domain in which they were defined. Domain attribute credentials are defined as SAML assertions, and are issued by each user. Only assertions regarding their own domain attributes can be issued by users of the attribute federation.

These type of attributes make the system more dynamic because they do not require the same verification process of Global Attributes, which could be a very expensive process.

In some cases, there are relationships between attributes, i.e. $GenA_Senior$ may imply $GenA_Junior$, as a senior assistant already has an ample experience managing a certain generator. Then, along with the attributes, there is a partial order that defines a simple ontology [**?**] in the domain of the attribute manager.

An ontology is a data model that represents a set of concepts within a domain and the relationships between those concepts. It is used to

reason about the objects within that domain. This ontology encodes the relationships between the attributes in the system. Moreover, there may be relationships between attributes in different domains, e.g. different Professors may issue different attributes. In order to make a reference to the attribute domain we use the dot notation. For instance, $GenA_Junior$ is an attribute created by Operator1 and managed by him which states that a certain assistant has completed the courses to know how to operate a Generator Class A. This attribute on its own has only a local meaning in the domain of Operator1.

In the Power plant scenario, the Power plant must provide a server to store the particular attributes defined by its members and also the relationships between them.

Now, let us suppose that the experience obtained by being taught by Operator1 is accepted by switchboard operators located in other power plants (e.g. Operator2) as a proof that this particular assistant can be able to manage a certain Generator Class A. In this case, Operator2 may rely on the previously defined attributes and define an attribute relationship stating that the "Junior" experience level implies that the assistant has experience in managing Generators Class A. In this way, Operator2 gives the previous attributes defined by Operator1 a new meaning, by uploading the following relation to the power plant server, using RT notation (cf. section B.2.1):

$$Operator1.GenA_Junior \rightarrow Operator2.GenA_Experience$$

This can be also represented using the partial order symbol,

$$Operator2.GenA_Experience \leq Operator1.GenA_Junior$$

In this case, Operator2 is delegating his authorization on the knowledge of Generator Class A systems to Operator1. Then, an assistant who is trying to convince the power plant managers where Operator2 resides about his experience with Generators Class A, may show his identity details to the power plant so it could ask Operator2. Or also, he may show Operator1's attribute stating that he has being taught by Operator1 on the operation of Generator Class A systems together with the attribute relationship

$$Operator2.GenA_Experience \leq Operator1.GenA_Junior$$

signed by Operator2. So, the process can be carried out without involving Operator2.

Although Domain and Global attributes are different concepts, they can also be related in the attribute ontology. In particular, any user will

be able to issue an attribute subscription of one of its domain attribute to one or more global attributes. By issuing this attribute subscription, this user is giving a local meaning to the global attributes inside his domain.

B.2.3 Attribute Federation: Components

The Attribute Federation has two elements, the Attribute Authority and the Attribute Ontology Server (AOS), which can be composed of many subordinated AOS. The AA manages Global Attributes whereas the AOS manages the Domain Attributes. Therefore, the main element that the attribute federation introduces, in contrast with traditional privilege management infrastructures, is the AOS, which is in charge of:

1. Managing Domain Attributes.
2. Storing relationships over Domain Attributes in the system.
3. Checking attribute relationships based on the stored ontology, i.e. performing the trust negotiation autonomously.

As we can see in figure 10, the existing infrastructure certification authorities issue public key certificates to each user in the system. This keys will be used later to sign SAML attribute statements. In order to include global attributes, i.e. those issued by AAs, into the federation we have to translate the X.509 Attribute Certificate into SAML assertions by using XSAML (see section B.2.1). All the SAML assertions passed to the AOS are used to update the respective ontologies. In figure 10 we can see the interrelations of each of the components of the attribute federation.

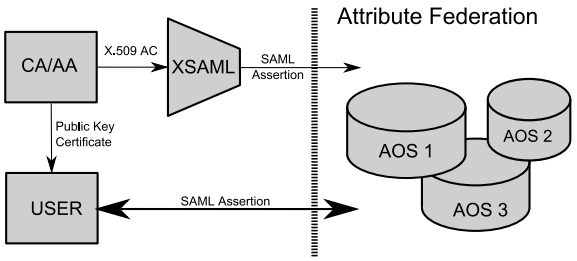


Fig. 10: Relation between the attribute federation's components

Every entity is registered to an AOS which is in charge of checking the relationships over attributes. Registered users trust the AOS not to disclose their attribute relationships neither to cheat them adding fake

relationships. All the communication with the attribute federation, i.e. the AOSs, is carried out by using SAML assertions.

Users may store new relationships in their AOS and/or check whether a relationship exists or not in any of the AOS of the Federation. The kind of relationships a user can upload to the AOS are of the form $User1.Attribute1 \leq User2.Attribute2$, where $User1$ is the ID of the user that is uploading the subscription. We call them attribute subscription and we read it as "attribute $User1.Attribute1$ is subscribed to attribute $User2.Attribute2$". Attribute subscriptions are transitive in the sense that if $Attr1$ is subscribed to $Attr2$ and $Attr2$ is subscribed to $Attr3$, then we can infer that $Attr1$ is subscribed to $Attr3$. Attribute subscriptions is an analogous concept of Li linked roles [**?**].

By using attribute subscriptions, users can rely on some other user attributes when defining their own authorization policies, but they can not force other users to rely on their own attributes. In those cases in which the attribute ontology has the be kept confidential, some restriction may apply when checking attribute subscriptions. For example, in order to preserve the privacy of the attribute ontology, a check for a relation of the form $User1.Attribute1 \leq User2.Attribute2$ is only answered to a user owning attribute $User2.Attribute2$.

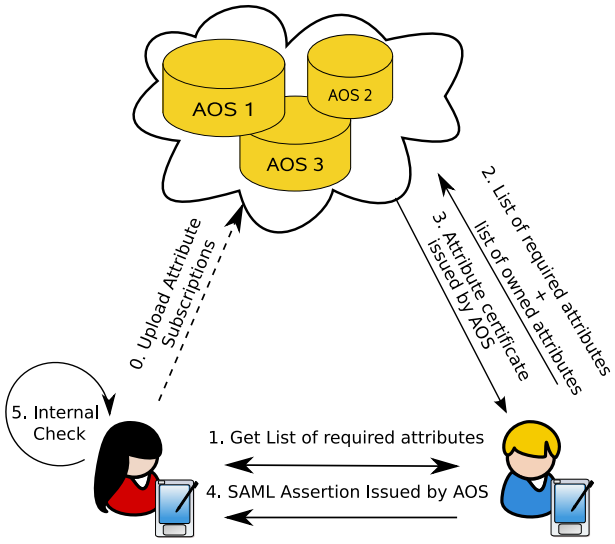


Fig. 11: Interaction Protocol

Let think of an scenario (cf. Figure 11) in which Alice (an operator) wants to contact Bob (another operator) in order to retrieve the attributes needed to use the service she wants to use (e.g. the management of a certain generator). This can be done contacting Bob directly or by checking some other public service used by Bob to publish his authorization policies, e.g. a static web page.

Once Alice knows the required attributes, she looks for attributes that may be related to her own. If the authorization policy is confidential, Alice will only get a specific domain attribute (in the domain of Bob) linked with the service she wants to access (e.g. $Bob.service1$), and the policy will remain confidentially in the AOS.

Then, she contacts the respective AOS in order to check whether there is a real relation between the candidate attributes. This checking is done by Alice, while Bob is not involved. Once Alice has found a relation between one of her attributes and one of the required attributes, she is ready to initiate a session with Bob in order to use the service. In this way, Bob is only slightly involved in the protocol because he only has to locally check the response of the AOS instead of having to check an attribute matching for all the users trying to use his web service.

B.3 Delegation and Quotas

When delegation takes place in the electronic world, the delegator usually retains also the privileges. Thus, both users have them simultaneously. This situation may be undesirable when dealing with finite resources, those in which the number of users accessing simultaneously is finite. In this section we provide an introduction of an approach where each user is delegated an access quota for a resource. If further delegating of the delegated quota occurs, this is subtracted from his quota. That is, when delegating, part of the quota remains with the delegator and another part goes to the delegatee. As for the underlying technology, we will make use of the extension mechanisms of PMIs.

B.3.1 Example Scenario

Let us assume a grid composed of different organizations managing various critical infrastructures which want to share computational resources. Each organization has a participating quota that determines the influence in the authorization process in order to use the resources.

When making authorization decisions this hierarchy, and the participation indexes of quota, have to be taken into account. There are several ways of doing this.

Each entity may issue different certificates, and those will be weighted according to their participation index.

Each entity participating in the grid has a share of quota. This share of quota could be handed over to different actors (e.g. users, infrastructures). Usually, each organization would keep some of this quota for its use and will 'sell' the surplus of resources. This process could take place in a cascade effect. That is, if a certain infrastructure has bought a participation in a particular resource, this infrastructure could also sell a portion to another external organization.

As consecutive sales advance, the quota that the new infrastructures obtain decreases. That means that more external infrastructures to the grid will have less influence on managing rights to access the resources of the grid.

There should also be a policy of access for each resource established by the grid and a common agreement where the minimum quota needed for accessing such a resource is reflected. Depending on the nature of the resource, this quota will be higher or lower. For instance, if the resource is a cluster, a request for a minute slot of use requires a lower quota than for using it for longer. Also, if there are two or more users competing for the same resource the quota can be used in order to prioritize the access to the resource.

B.3.2 Quota Approach for Delegation

In the previous section we have presented a scenario where the entities participating in a grid (i.e. organizations managing certain infrastructures) delegate rights management to another organizations. In this section we will introduce a mathematical model that formalizes the situation of the scenarios described above. We will call this model the *Quota Delegation Model*.

Since organizations in lower levels have been delegated less quota as we descend one level in the chain, we are interested in the exact quota that any organization retains. The quota is a real number in the interval $[0, 1]$, representing a percentage of the total quota where 1 corresponds to 100%. A user expresses the quota delegated to other entities relatively to the actual quota they obtain. The actual quota that a user delegates to another one is computed by multiplying the relative quota (encoded in the delegation quota credential) and the actual quota of the delegator. The absolute quota of a user is computed by summing up all the absolute quotas delegated by all the users of the system. In this section we will

provide with an iterative mechanism for computing the absolute quota of each user.

In order to compute the quota of a certain organization we will use the product of all the quota of the links in the chain from the root node to the node we want to compute, minus the quota that this entity delegates. In fact, for each entity we can distinguish the delegated quota, which is the quota that reaches this entity trough delegation paths, and the actual quota of this entity that is computed by subtracting the quota delegated to other entities from the delegated quota of this entity.

If there are several chains for delegating quota to organizations, the final quota will be the addition of all the quotas obtained from all the different chains.

Loops are not allowed in our quota delegation model and have to be solved outside of it. For instance, if entity X has delegated some of its quota to entity Y and later, Y wants to delegate some of its quota to X, then there are two possibilities:

- If the absolute quota that Y wants to delegate to X is greater than the absolute quota X delegates to Y, then the quota delegation credential from X to Y should be removed from the system and a new quota delegation credential from Y to X should be added, where the delegated quota corresponds to the difference between the absolute quota that Y wants to delegate to X and the absolute quota delegated from X to Y expressed relatively to the quota of Y.
- If the absolute quota that Y wants to delegate to X is lower than the absolute quota X delegates to Y, then the current quota delegation credential from X to Y should be removed from the system and a new quota delegation credential from X to Y should be added. Then the delegated quota corresponds to the difference between the absolute quota delegated from X to Y and the absolute quota that Y wants to delegate to X expressed relatively to the quota of X.

In both cases the resulting quota delegation graph has no loops.

We have initially used attribute certificate credentials [**?**,**?**] to implement our Quota Delegation Model. The privilege delegated is encoded as an attribute and the value of the attribute is set to the quota. All the credentials have to be passed to a central server which checks that the quota assignments that each user does for each privilege is fair, i.e. the quotas of each credential issued by the same user and regarding the same privilege sum less than 1, which represents 100% of the quota. This central server stores all the delegated quotas in a matrix form.

If we establish an equivalence between nodes or organizations and states, and between the quota that an entity X hands over to Z and the statistical concept of transition from one X to Z, our particular problem could be modelled as a discrete Markov's chain where the number of phases or stages corresponds to the length of the chain that we consider.

Computational Model. The initial organization or entity that first holds all the quota is called the *initiator* and it will be the initial state of the Markov's chain. The values of the quotas could be placed in a matrix such as the addition of the elements in each row is lower or equal than 1 ('almost' a stochastic matrix). The reason is that the addition of all quota could never be greater than 1 but it can be less, i.e., there could be states without any assigned quota. Therefore, in order to make this matrix a stochastic matrix we should add an additional state, namely, the state of the non-assigned quota. This new state will mean a new column and row for the matrix of the states where the values in the columns are calculated in such a way that the addition of the values in the row is 1. If we call this matrix A, the associated stochastic matrix A^* is as follows:

$$\left(\begin{array}{c|c} & 1-\sum_{j=1}^{n} a_{1,j} \\ A & \vdots \\ & 1-\sum_{j=1}^{n} a_{n,j} \\ \hline 0\cdots 0 & 1 \end{array}\right)$$

The state of 'non-assigned' quota is a non-transition state as once it has been reached no other state can be reached afterwards.

This matrix can be expressed in its canonical form as

$$\left(\begin{array}{c|c} A & R \\ \hline 0 & I \end{array}\right)$$

where A is the matrix of quota delegation that contains all the transition states; R is the matrix that contains the non-assigned quota by entities and I is the identity matrix of length 1. As there are no loops in the quota delegation graph, the matrix A is upper triangular or at least we can make it upper triangular by reordering the nodes using a topological sort algorithm over the quota delegation graph. Therefore, the matrix of the chain, N, is calculated as follows:

$$N = (I - A)^{-1}$$

Next we will show that the element n_{ij} of N is the share of quota that entity i hands over to entity j.

An element $a_{ij}^{(k)}$ of matrix A^k represents the percentage of quota that node i hands over node j indirectly, if we only consider chains of length k. Also, $A^n = 0$ if n is greater or equal than the size of the matrix, as this is an upper triangular matrix. Therefore, we can define matrix $\widehat{A}$ as

$$\widehat{A} = \sum_{i=1}^{\infty} A^i = \sum_{i=1}^{n-1} A^i$$

The elements of $\widehat{A}$ can be calculated in the following way:

$$\hat{a}_{ij} = \sum_{s=1}^{n-1} a_{ij}^{(s)}$$

$$\widehat{A} = (\hat{a}_{ij})$$

The elements $\hat{a}_{ij}$ of $\widehat{A}$ are the addition of all the quota that node i hands over to node j for chains of any length.

Next we will show that $N = I + \widehat{A}$, i.e., $I + \widehat{A}$ is the inverse of $I - A$. In order to do this, we will show that the matrix is invertible from the right-hand side (it is analogous from the left-hand side).

$$\begin{aligned}
(I - A)(I + \widehat{A}) &= (I - A)(\sum_{s=0}^{n-1} A^s) \\
&= \sum_{s=0}^{n-1} A^s - \sum_{s=1}^{n} A^s \\
&= I + \sum_{s=1}^{n-1} A^s - \sum_{s=1}^{n-1} A^s - A^n \\
&= I
\end{aligned}$$

This gives us two ways of obtaining the percentage of quota handed over by entity i to entity j. Either by calculating the element ij of matrix N or by calculating the element (i, j) of matrix $\widehat{A}$.

In both cases, we can use the first row of those columns to form the vector of quota delegation from the initiator.

Definition 1 (quota delegation vector). *The quota delegation vector is the vector v consisting of all the quota delegation values from the initiator to the rest of the entities. The first element is set to* 1.

In order to obtain the actual quota that remains in each entity, we have to subtract the quota it delegates from the quota it has been delegated. This can be easily done by multiplying the corresponding element of the quota delegation vector, v_i, by the element r_i of the column vector R.

By multiplying the column vector R by v, element by element, we obtain the quota distribution over the entities. The sum of all the elements of this vector is 1.

B.3.3 Quota Delegation Model and Example

Our quota based delegation model is mainly focused on facilitating delegation of access to resources that can be measured and consequently divided among users according to the specified quota percentage. An example of such a kind of resources is the CPU load in cluster environments. In this case there is not need for a specific authorization policy to be used against the certificates, as they encode both things. The rights are already included in the credential, therefore we do not need to contrast it with an authorization policy. However, not all the resources are easily split and furthermore, sometimes it is undesirable to include the resource in the credential. In those cases, we use a role or group membership attribute and split this attribute among the users in the system. The grid scenario is a clear example of this situation. In those cases, we do need an authorization policy, therefore actual privileges or rights can be derived from the quota membership to a particular attribute. In Figure 12 we illustrate how two scenarios (the previously mentioned grid scenario and a residential network scenario where resident shares resources) are characterized according to where more effort is needed, either in the definition of credentials or in the definition of the authorization policies.

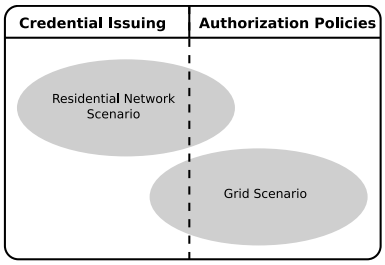


Fig. 12: Characterization of the two scenarios

Even though we have made a distinction here between these two examples, there might be cases where it is not that easy to make it. For the definition of the authorization policy we take as an input the quota or the percentage of the attribute that the user holds, and the higher it is, the more privileges it will be delegated.

The authorization problem can be also tackled by using negative statements, such as 'this user will never access this resource'. As each user is delegated a quota of the resource, an authorization credential can be given a weight associated to the quota of the issuer. Then, positive authorization credentials, i.e., those granting some privileges but not delegating them, can be counted as positive votes for authorization decisions and negative ones as negative votes. Those votes are proportional to the quota of the issuers of the credentials, in such a way that at the end, we can sum up all the positive votes and subtract from them all the negative ones. If the result is positive the authorization request will be granted. However, this is just a simple authorization policy. More complex solutions can be defined, such as requesting a lower bound for the delegated quota.

Example. Let us assume a grid of four organizations X, Z, V and W. Let us also assume that the X is the user who initially possesses 100% of the quota of a given resource, i.e. X is the owner of the resource and wants to contribute a share of it to the grid. X hands over a third of the quota to V and another third to W. V and W also delegate 3/4 of their quota to Z. We are interested in calculating the exact quota that each entity in the grid retains after all the quota delegations are effective.

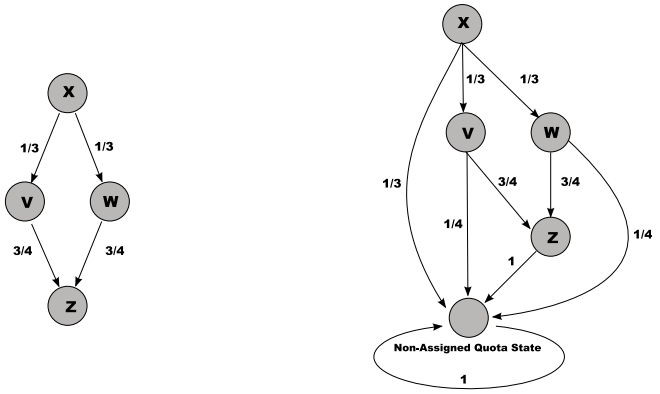
In order for the matrix A to be upper triangular, we can establish the following order of the nodes: $X \to 1$, $V \to 2$, $W \to 3$ and $Z \to 4$.

Figure 13 shows the distribution of the quota. The Figure on the left-hand side describes how the quota originally is distributed among peers and, on the one on the right-hand side the distribution of quota is represented as a States Transition Diagram (STD) of the associated Markov's chain which includes the 'non-assigned' quota states.

The matrix representing the quota assigned is as follows

$$A = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{3}{4} \\ 0 & 0 & 0 & \frac{3}{4} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

From this matrix we can obtain the stochastic matrix by including the non-assigned quota in the last row and column.



(a) Original Assignment of quota

(b) STD of the associated Markov's chain

Fig. 13: From quota assignment graph to Markov's chain STD

$$A^* = \left(\begin{array}{cccc|c} 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{3}{4} & \frac{1}{4} \\ 0 & 0 & 0 & \frac{3}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

The fifth row and column correspond to the state of non-assigned quota.

In order to calculate the quota that X hands over Z we should calculate, in Markov's processes terminology, the probability of going from state X to Z. We will use matrix N for doing it.

$$N = (I - A)^{-1} = \left(\begin{array}{cccc} 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} \\ 0 & 1 & 0 & \frac{3}{4} \\ 0 & 0 & 1 & \frac{3}{4} \\ 0 & 0 & 0 & 1 \end{array}\right)$$

In order to obtain the actual quota belonging to each user, we have to subtract the quota that has already been delegated. We can do this by multiplying the first row of this matrix by the column of the non-assigned quota, element by element.

$$(1, \frac{1}{3}, \frac{1}{3}, \frac{1}{2}) \times (\frac{1}{3}, \frac{1}{4}, \frac{1}{4}, 1) = (\frac{1}{3}, \frac{1}{12}, \frac{1}{12}, \frac{1}{2})$$

Therefore, the shares of assigned quota of all the participants are the corresponding elements of this vector. Note that the sum of the elements of this vector is one, therefore the quota property holds.

If we implement an authorization policy such as the simple one defined in the previous section in such a way that Z and W decide that access to a third party has to be granted, it does not matter what X and Y state, as the votes of Z and W count more than 50%. Thus, the decision will be to grant access.

References

1. Agudo, I., Lopez, J., Montenegro, J.A.: A Representation Model of Trust Relationships with Delegation Extensions. In: 3th International Conference on Trust Management (iTRUST'05). LNCS, vol. 3477, pp. 116–130. Springer, Springer, Versailles, France (May 2005)
2. Bianchi, G., Capossele, A.T., Mei, A., Petrioli, C.: Flexible key exchange negotiation for wireless sensor networks. In: Proceedings of the fifth ACM international workshop on Wireless network testbeds, experimental evaluation and characterization. pp. 55–62. WiNTECH '10, ACM, New York, NY, USA (2010)
3. Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.: RFC 2704: The KeyNote Trust Management System Version 2 (1999)
4. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. In: IEEE Symposium on Security and Privacy. pp. 164–173 (1996)
5. Bray, T., Paoli, J., Sperberg-McQueen, C., Maeler, E., Yergeau, F.: Extensible Markup Language (XML) 1.0 (Fifth Edition) (November 2008), w3C Recommendation. World Wide Web Consortium (W3C)
6. Christin, D., Reinhardt, A., Mogre, P., Steinmetz, R.: Wireless Sensor Networks and the Internet of Things: Selected Challenges. Proceedings of the 8th GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze", 2009
7. DAML+OIL web ontology language: http://www.w3.org/TR/daml+oil-reference, Accessed on October, 2011
8. Dierks, T., Rescorla, E.: RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2 (2008)
9. Ellison, C.: RFC 2692: SPKI Requirements (1999)
10. Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., Ylonen, T.: RFC 2693: SPKI Certificate Theory (1999)
11. Eronen, P., Tschofenig, H.: RFC 4279: Pre-Shared Key Ciphersuites for Transport Layer Security (TLS) (2005)
12. Farrell, S., Housley, R.: RFC 3281: An Internet Attribute Certificate Profile for Authorization (2002)
13. Forne, J., Hinarejos, F., Marin, A., Almenarez, F., Lopez, J., Montenegro, J.A., Lacoste, M., Diaz, D.: Pervasive authentication and authorization infrastructures for mobile users. Computers and Security 29(4), 501–514 (2010)
14. Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies 43(5–6), 907–928 (1995)
15. Gungor, V., Lambert, F.: A survey on communication networks for electric system automation. ACM Computer Networks 50(7), 877–897 (2006)

16. Hinarejos, M.F., Muñoz, J.L., Forne, J., Esparza, O.: Preon: An efficient cascade revocation mechanism for delegation paths. Computers and Security 29(6), 697–711 (2010)
17. IBM Corporation, A Strategic Approach to Protecting SCADA and Process Control Systems: http://www.iss.net/documents/whitepapers/SCADA.pdf, Accessed on October 2011
18. ITU-T: Recommentation X.509: The Directory–Authentication Framework (1997)
19. ITU-T: Recommentation X.509: The Directory–Public-key and attribute certificate frameworks (2000)
20. Jain, M., Jain, A., Srinivas, M.: A web based expert system shell for fault diagnosis and control of power system equipment. In: Condition Monitoring and Diagnosis (CMD 2008). pp. 1310–1313 (2008)
21. Johnson, M., Healy, M., van de Ven, P., Hayes, M., Nelson, J., Newe, T., Lewis, E.: A Comparative Review of Wireless Sensor Network Mote Technologies. In: IEEE Sensors 2009. pp. 1439–1442 (October 2009)
22. Kushalnagar, N., Montenegro, G., Schumacher, C.: RFC 4919: IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals (2007)
23. Leou, R., Chang, Y., Teng, J.: A Web-based Power Quality Monitoring System. In: IEEE Power Engineering Society Summer Meeting (Vol. 3). pp. 1504–1508 (2001)
24. Li, D., Serizawa, Y., Kiuchi, M.: Concept design for a Web-based supervisory control and data-acquisition (SCADA) system. In: IEEE Transmission and Distribution Conference and Exhibition, Asia Pacific (Vol. 1). pp. 32–36 (2002)
25. Li, N., Mitchell, J.C., Winsborough, W.H.: Design of a role-based trust management framework. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy. pp. 114–130 (2002)
26. Liu, A., Ning, P.: TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In: International Conference on Information Processing in Sensor Networks (IPSN 2008). pp. 245–256 (April 2008)
27. Lopez, J., Alcaraz, C., Roman, R.: On the Protection and Technologies of Critical Information Infrastructures. In: On Foundations of Security Analysis and Design IV, FOSAD 2006/2007, Springer, LNCS, vol. 4677, pp. 160–182 (2007)
28. Lopez, J., Montenegro, J.A., Roman, R.: Service-oriented security architecture for cii based on sensor networks. In: 2nd International Workshop on Security Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU'06). pp. 1–6. IEEE Press, IEEE Press, Lyon, France (June 2006)
29. Malan, D., Welsh, M., Smith, M.: A Public-key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography. In: 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON 2004). pp. 71–80 (October 2004)
30. Montenegro, G., Kushalnagar, N., Hui, J., Culler, D.: RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks (2007)
31. Mundy, D., Chadwick, D.: An XML alternative for perfomance and security: ASN.1. IEEE IT Professional 6(1), 30–36 (2004)
32. Neuman, C., Yu, T., S. Hartman, a.K.R.: RFC 4129: The Kerberos Network Authentication Service (V5), Request for Comments (2005)
33. Nivis VN210 sensor node datasheet: http://www.nivis.com/industrial_sensor_networks/VersaNode.php, Accessed on October 2011
34. OASIS Web Services Secure Exchange (WS-SX) TC: http://www.oasis-open.org/committees/ws-sx/, Accessed on October 2010

35. Oliveira, L.B., Aranha, D.F., Gouvea, C.P., Scott, M., Camara, D.F., Lopez, J., Dahab, R.: TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. Computer Communications In Press, Corrected Proof, – (2010)
36. Qiu, B., Gooi, B.: Web-based SCADA display systems (WSDS) for access via Internet. IEEE Transactions on Power Systems 5(2), 681–686 (2000)
37. Qiu, B., Gooi, H., Liu, Y., Chan, E.: Internet-based SCADA display system. Computer Applications in Power 15(1), 14–19 (2002)
38. Rabin, M.: Digitalized signatures and public-key functions as intractable as factorization. Tech. rep., Cambridge, MA, USA (1979)
39. Rescorla, E., Modadugu, N.: RFC 4347: Datagram Transport Layer Security (2006)
40. Rivest, R., Lampson, B.: SDSI - A Simple Distributed Security Infrastructure. In: 6th Annual International Cryptology Conference (1996)
41. Roman, R., Fernandez-Gago, M.C., Lopez, J., Chen, H.H.: Trust and reputation systems for wireless sensor networks. In: Security and Privacy in Mobile and Wireless Networking, pp. 105–128. Troubador Publishing Ltd (2009)
42. Roman, R., Lopez, J.: Integrating wireless sensor networks and the internet: A security analysis. Internet Research 19(2), 246–259 (2009)
43. Smith, M.: Web-based Monitoring & Control for OilGas Industry. Pipeline & Gas Journal (2001)
44. WebSCADA: http://www.webscada.com/, Accessed on October 2011
45. X.690, I.R.: ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) (1995)
46. Yokogawa: http://yokogawa.com/scd/fasttools/scd-scada-websuper-en.htm, Accessed on October 2011
47. ZigBee Alliance: http://www.zigbee.org/ (2009)