FP7-ICT-STREP

Contract No. 258280

TWISNet

Trustworthy Wireless Industrial Sensor Networks

Deliverable D3.3.2

Specification of security services for multi-owner scenarios

Contractual date:	M30
Actual submission date:	M30
Responsible beneficiary:	UPB
Authors:	Dr. Alexis Olivereau (CEA), Andrei Voinescu (UPB), Basil Hess (SAP), Dr. Dan Tudose (UPB), Dr. Emil Slusanschi (UPB), Dr. Felix von Reischach (SAP), Dr. Laura Gheorghe (UPB), Dr. NouhaOualha (CEA).
Work package:	WP3
Security:	CO
Nature:	Report
Document name:	TWISNet_WP3_D3.3.2.doc
Version:	2.0

REVISION HISTORY

Date (dd.mm.yyyy)	Version	Author	Comments
18.02.2011	1.0	Emil Slusanschi	ToC for chapter 1 created
20.04.2011	1.1	Basil Hess	Added Sota part of SAP
22.04.2011	2.0	Emil Slusanschi	Added Sota part of UPB
25.04.2011	3.0	Dan Tudose	Updated Sota part of UPB
12.05.2011	4.0	Alexis Olivereau	AddedSota part of CEA
01.06.2011	5.0	Emil Slusanschi	Added conclusions to chapter 1
20.07.2011	6.0	Emil Slusanschi	ToC structure for chapter 2 added
10.10.2011	7.0	Alexis Olivereau	Added chapter 2 part of CEA
20.11.2011	8.0	Laura Gheorghe	Addedsection 2.2
15.12.2011	8.1	Dan Tudose	Updated section 2.3
30.01.2012	9.0	Emil Slusanschi	ToC structure for chapter 3 added
20.02.2012	9.1	Alexis Olivereau	Added section 3.1
23.02.2012	9.2	Dan Tudose	Added section 3.4
23.02.2012	9.3	NouhaOualha	Added sections 3.2 and 3.3
22.03.2012	9.4	Andrei Voinescu	Updated section 3.4
27.03.2012	9.5	Alexis Olivereau	Updated section 3.1
28.03.2012	9.6	NouhaOualha	Updated sections 3.2 and 3.3
29.03.2012	10.0	Emil Slusanschi	Review of chapters 1, 2, 3
03.04.2012	11.0	Markus Wehner	Complete review
09.04.2012	12.0	Emil Slusanschi	Address review comments
05.03.2013	1.0	Laura Gheorghe	ToC for D3.3.2 based on D3.3.1
25.03.2013	1.1	Emil Slusanschi	Updated sections 1, 2, 3, 4, 5.
27.03.2013	1.2	Andrei Voinescu	Updated section 4.4
03.04.2013	1.3	Nouha Oualha	Updated subsections 4.2 and 4.3
09.04.2013	1.4	Aymen Boudguiga	First review completed
10.04.2013	1.5	Alexis Olivereau	Privacy Manager section updated
11.04.2013	1.6	Aymen Boudguiga	Privacy Manager section reviewed
11.04.2013	1.7	Nouha Oualha	Addressed comments related to selective delivery and connection sharing.
11.04.2013	1.8	Andrei Voinescu	Reworded paragraphs in 3.3.4.1 in response to comments.
11.04.2013	2.0	Emil Slusanschi	Final document review.

Abstract:

Wireless Sensor Networks have drawn the attention of the research community fordecades. They provide applications in monitoring and controlling of industrial processes,workers, environmental phenomena, and many more. However, because sensors typicallycommunicate over the air, there is always a certain danger that information can beaccessed and modified by unauthorized parties. TWISNet (Trustworthy Wireless IndustrialSensor Networks) aims at addressing security issues arising when applying wirelesssensor networks to industrial environments.

This document will detail the way in which the TWISNet project aims to tackle the issues of privacy management, selective delivery, connection sharing, and sensor co-management. Starting from the state of the art we propose a detailed architecture and specify the technical modules which will focus these security issues.

The current state of the art in security services has been examined with respect to collaborative projects, existing standards and available commercial products. An integrated architecture for each of these topics of research is presented, and a detailed module technical description is given, detailing tasks, actions and the envisaged protocol. The proposed architecture and corresponding modules will be refined during the implementation and integration of these technologies in WP3 and WP4 respectively. All these issues will be presented in the final version of the deliverable.

Keyword list:

Security requirements, privacy management, selective delivery, secure connection sharing, sensor co-management, data privacy, identity privacy

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
AAA	Authentication, Authorization, Accounting
ACM	Access Control Manager
AES	Advanced Encryption Standard
DAP	Data Aggregation and Processing
FP6/FP7	Sixth/Seventh Framework Programme
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ID-FF	Identity Federation Framework
IT	Information Technology
J2ME	Java 2 Platform, Micro Edition
KEK	Key Encryption Keys
KM	Key Management
MAC	Medium Access Control; Message Authentication Code
MSN	Medical Sensor Networks
P2P	Peer-to-Peer
PAN	Personal Area Network
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PGP	Pretty Good Privacy
PKC	Public Key Certificate
PKIs	Public Key Infrastructures
PTM	Pervasive Trust Management
RBAC	Role-Based Access Control
SAML	Security Assertion Markup Language
SCM	Sensor Co-Management
SPKI	Simple Public Key Infrastructure
SSO	Single Sign-On
TEK	Traffic Encryption Keys
TGK	TEK Generation Key
TTP	Trusted Third Parties

UPnP	Universal Plug'n'Play
URI	Uniform Resource Identifier
VM	Virtual Machine
WSAN	Wireless Sensor and Actor Network
WSN	Wireless Sensor Network
XRI	Extensible Resource Identifier

1. INTRODUCTION

This deliverable contains the outcome of activities taking place in task 3.3 of the TWISNet project, and it addresses relevant issues for security services in multi-owner scenarios, such as privacy management, selective delivery, connection sharing, and sensor co-management topics.

For each of these topics, the current state of the art is examined in Chapter 2. Also, relevant collaborative projects, existing standards and available commercial products are summarized for each of these issues. Chapter 2 concludes with a summary of the state of the art in security services.

Chapter 3 proposes a detailed integrated architecture for all the identified technical modules tackling the various security issues. The modules developed in task 3.3 are Privacy Manager, Selected Delivery, Connection Sharing and Sensor Co-Management. The interactions among these modules within the TWISNet framework are shown in the proposed architecture.

Chapter 4 details the tasks, actions as well as the envisaged protocols which are implemented in each of the modules composing the integrated architecture described in Chapter 3.

The deliverable concludes in Chapter 5 with a brief summary and outlook on the issues that are still under development.

2. STATE OF THE ART

2.1 Research literature

2.1.1 Privacy Manager

In the field of cloud computing, [1] proposes aclient-based privacy manager aiming at making sure that no private information leaks to the cloud without user consent. The so-called privacy manager component is actually the interface between user's device and cloud application. It is used to perform data obfuscation, preference setting, data access (audit of what information is held by the cloud regarding a certain user), feedback (client-side feedback to the user regarding usage of his personal information) and personae (virtual identifiers) control.

In [2], a privacy management system is defined that consists in a middleware framework on a middleware platform, interfacing with clients' applications. This middleware intercepts personal information requests and enables a privacy notification service, which can work either in pull mode (privacy aware application regularly polls the platform) or in push mode. A notified application may eventually notify the user himself about privacy concerns. Pull mode is more resource-consuming, but also more predictable.

The solution proposed in [3] consists of a system of rules regarding the disclosure of private information. These rules state that an access attempt to private information will lead to a*Permit*, *Reject*, or *Question* decision. In the latter situation, the user has to be informed about the reason why her private information has to be used. Her explicit and cryptographically protected consent has then to be obtained. A dedicated user consent protocol is proposed to achieve the user consent gathering.

2.1.2 Controlled access to user data

Sensor data generally requires a suitable access control in order to avoid unauthorized access or modification of relevant information. Several access control mechanisms have been devised for WSNs.

The system proposed in[4]allows a modular access control for Medical Sensor Networks (MSNs). The system is able to make context-aware and fine-grained decisions through three levels: a highly acute or critical access control level in which the need for user authentication is removed, an emergency access control level where authentication is required but privacy is traded-off with safety, and a normal context-aware access control level. Taking decisions based on these three levels renders fast the access control decisions and the response depending upon the health acuteness of the user. Moreover, access control decisions take place in-network since part of policy modules are distributed to sensor nodes, which is of great interest for a system limited in bandwidth and with intermittent connectivity.

The system in [5] proposes to enforce access control over user data through data encryption. In the proposed system, each data type is mapped to an authorization level associated with asecret key. The knowledge of such key used for data encryption will allow access to such data. Thus, the problem of access rights management is reduced to the problem of key management. In the proposed system, the keys used to access data are organized in a hierarchical structure whereby low-level keys can be derived from high-level ones. Hash-based tree construction has been proposed to build keys at the different levels of the hierarchy. The tree root key of the hierarchy is stored by sensor nodes. It is updated whenever access rights of a user have been revoked. Moreover, the system handles the case where a node is compromised as a key revocation case by updating key materials. So, for user access revocation or to handle node being compromised, the whole key hierarchy

should be updated using broadcasted messages from a centralized authority, called ACM (Access Control Manager); this is not practical if some sensor nodes have limited connectivity (i.e., access rights revocation can be partial).

2.1.3 Sensor co-management

Sensor co-management in WSAN implies the existence of some sort of AAA (Authentication, Authorization, Accounting) in the underlining architecture whereboth user and device must be authenticated.

Access Control is an approach to restrict access to system resources to authorized users. Access control techniques can roughly be categorized into the following models:

- Discretionary access control lets the owner of an object determine the access policy. A subject who owns an object may assign or revoke access rights to this object for him and other subjects. The standard access control model for discretionary access control is by Harrison et al.[6].
- Mandatory access control has a system wide fixed access policy. Every subject has a clearance and every object a classification level. To access an object, the subject must have a clearance at least as high as object's classification level. The standard modelis the Bell-LaPadula model [7].
- Role-based access control is a newer approach [8]. In a role-based access control, the access policy is not defined for subjects, but on roles that are assigned to the subjects.

Context-based access control mechanisms [9] are particularly interesting for WSANs. They are shaped for a ubiquitous computing infrastructure and allow access decisions to depend on the context in which requests are made. This is done by extending the role concept to capture security relevant context of the environment in which access requests are made.

Authentication and authorization frameworks are architectures for providing and managing access control. They provide methods for controlling the identification and authentication of users and for administering which authenticated users are granted access to protected resources. Some of the frameworks described can be used to provide several functions as shown in the table below.

Framework	**Identity Certificate Management**	**Single Sign-On**	**Federated Identity**	**Policy-based Security Management**	**User-centric**
PKI	X				
PGP	X				
Kerberos		X			
Windows Live ID		X			X
OpenID		X			X
LibertyAlliance		X	X		X
WS-Federation		X	X		
IETF Policy				X	

Identity certificate frameworks allow users without prior contact to authenticate to each other and digitally sign and encrypt messages. They are based on identity certificates, which are certificates that bind a public key to an identity. These frameworks are used to prove that the public key is authentic (i.e. belongs to the claimed identity) and that it has not been tampered with. Examples of identity certificate frameworks include:

Public Key Infrastructures (PKIs), described in section **Fehler! Verweisquelle konnte nicht gefunden werden.**, such as X.509[10], which is an ITU-T standard for PKI and PKIX[11], which is a public key infrastructure based on the X.509 standard.

PGP (Pretty Good Privacy) [12]uses a combination of public key and symmetric key encryption to provide security services such as digital signatures and confidentiality for data files and email messages. It uses identity certificates to distribute public keys and certificates are validated through a 'web of trust'. This concept means that an identity certificate may be digitally signed by a third party user to certify the association between a user name and the key.

Single sign-on (SSO) allows users to be authenticated only once in a system. Users can then access all resources for which they have access permission without entering multiple passwords. Example SSO frameworks include:

Kerberos[13]: a distributed authentication service, which provides SSO within a single administrative domain. Initially, a user authenticates to the Authentication Server using shared secret key cryptography. A "ticket-granting" service [14] allows subsequent authentication without re-entering a password. Although a ticket cannot be revoked, it is time stamped to limit its lifetime.

Windows Live ID[15]: an Internet-based SSO framework used by Microsoft applications and web services such as MSN messenger. A central database stores account information, registration and sign-in/sign-out pages. Users initially authenticate using a username and password and are then issued an authentication token, which is used to authenticate to other services. These authentication tokens are removed from the system when the user logs out and have a limited lifetime so they are not stored infinitely on a user's computer.

OpenID[16]: an authentication framework that allows users to login to different web sites using a single digital identity, eliminating the need to have different usernames and passwords for each site. The digital identity can be any URI (Uniform Resource Identifier) (e.g. blog) or XRI (Extensible Resource Identifier) the user has control of. A user can change its OpenIDprovider by updating the discovery information available with their digital identity.

Federated Identity[17] allows users of one security domain to securely access resources on another security domain, without the need for another user account. Users register with an authentication server in their own domain and other domains trust its assertions. Examples include:

Liberty Alliance[18]: a consortium that aims to establish open standards, guidelines and best practices for federated identity management. It has produced a comprehensive set of specifications[19], which includes the Identity Federation Framework (ID-FF), describing the SSO and identity federation part of the framework.

WS-Federation[20]: a federated identity standard developed by Microsoft, IBM, VeriSign, BEA and RSA Security, which forms part of the Web Services Security framework[21]. It relies on a security token service and is an extension of that in WS-Trust[22]. It aims to provide a common infrastructure for performing Federated Identity operations for both web services and browser-based applications. WS-Federation only provides a framework for AAA; it does not provide implementations.

Policy-based management allows authorization to be partially automated. In a network with a large number of transactions, this is desirable as it reduces the amount of administration required. It separates policies from the implementation, which means that policies can be changed without affecting the underlying implementation. However, there is no agreed standard for policy-based security management, which affects the interoperability. It can also be difficult to implement as it has a more complex architecture than the other frameworks.

Mechanisms for implementing policy-based security management include:

The IETF Framework for policy-based Admission Control[23], which consists of a Policy Decision Point (PDP), which is responsible for handling events and making decisions based on events, a Policy Enforcement Point (PEP), which enforces the policy based on rules received from the PDP, a policy repository for the storage or retrieval of policy information and a user interface for specifying, administering and editing policies. These components can be co-located or distributed. There are a number of protocols that can be used for communication between the components, including Diameter.

Policy languages, which are used for specifying policies to enable policy-based management. They enable interoperability and ensure unambiguous mapping of a request profile to a policy action. Policy languages can specify the sequence in which different policy rules should be applied and/or the priority associated with each one. Leading examples include *XACML*[24] and *Ponder*[25].

The technologies that can be used to implement the authentication and authorization frameworks for policy-based management include certificate and token management and network access control protocols.

Certificates and tokens are digital documents that can be used for storing authorization information and other policies. They are used to transport policy information within and between domains. They are similar to public key certificates but provide different bindings and offer more flexibility than a Public Key Certificate (PKC). Examples of certificates and tokens include:

Attribute certificates[26]: structures similar to PKCs but bind an identity to attributes rather than to a public key. Identity certificates must be used with attribute certificates in order to allow users to prove ownership of them.

Authorization certificates bind authorization policies or assertions to public keys. They allow more general access control policies to be represented and do not require a naming infrastructure and so identity certificates are not needed. Example formats include SAML and SPKI (Simple Public Key Infrastructure)[27].

Proxy certificates [28]: allow the delegation of privileges by allowing the certificate issuer to be identified by a public key certificate or another proxy certificate instead of a CA certificate. Proxy certificates can therefore be created dynamically without requiring the normally heavyweight vetting process associated with obtaining PKCs from a CA. An example is the Grid Security Infrastructure[29].

Policy tokens [30]: structures used to specify a wide variety of security policies. They are signed by the issuer and are reissued every time a policy is changed. An example is KeyNote[31].

Network access control protocols enable centralized management of authentication and authorization. They allow the network access server to offload user administration to a central server. Access information is stored in a central access control list. Currently, network access control protocols are used for large networks but they may be usable for other systems or applications.

The first network access control protocols were RADIUS[32] and TACACS[33].Diameter [34] is the replacement for RADIUS. The Diameter base protocol is designed to provide a framework for applications relying on AAA.

User-centric identity management, also referred to as Identity 2.0, is a design principle that focuses on usability and cost-effectiveness from the user's point of view. It is similar to previous approaches such as SSO but it is mainly designed for individual users not for use in enterprises. It provides a consistent user experience, as a specific user employs the same identity agent for every identity transaction and separates identity component from rest of application. There are three main approaches to user-centric identity management:

- Managing multiple identities, e.g. information cards[35],[36];
- Giving users a single identity, e.g. OpenID;
- Giving users control over access to their resources, e.g. OAuth[37] .

2.1.4 Secure connection sharing

Connection sharing between sensor networks that possibly have different owners has to take into account the following basic considerationsfrom the literature[38][39]:

Connectivity: There is the need for sharing connectivity. This is an especial challenge since WSN usually use proprietary low-level communication protocols.

For enabling full connectivity, WSN traffic could be mapped on the low-level to IP networks. Exposing IP networks to LAN networks requires mapping security policies to the LAN network. A sample solution would include secured VPN channels.

Heterogeneity: Different communication protocols imply different encodings and formats. This heterogeneity poses a challenge for interconnecting the networks. For connection sharing, thereis the need for standardized abstractions.

The authors in [38] propose to use a service-oriented abstraction to the WSN. This enables uniformly accessibility of WSN resources across heterogeneous infrastructures.

Interoperability: To ensure that heterogeneous networks interoperate seamlessly, there is the need for a commonly defined overlay networks to enable ad-hoc connectivity.

For bridging the networks, [38]proposes to use especially powerful gateway nodes. 6LoWPAN[40]could be used to avoid dedicated gateway nodes. A protocol that uses direct IP connection is proposed in u-IP [41]. Here each sensor node is directlyaccessed over IP. In [**Fehler! Verweisquelle konnte nicht gefunden werden.**], indirect interconnection is employed by abstracting the access point to the WSN as an additional, virtual sensor node.

2.2 Collaborative projects

2.2.1 Controlled access to user data

UBISEC[42](Ubiquitous Networks with a Secure Provision of Services, Access, and Content Delivery) is an FP6 European-led project that aims to provide a mobile and secure infrastructure that provides context-aware and personalized authorization and authentication services. For access control, UBISEC proposes a distributed and autonomic solution, called TrustAC[43]that relies on a pervasive trust management (PTM) model for making decisions.TrustAC is a context-aware access control system that can work in open and dynamic environments, as it does not define pre-configured and fixed permissions for each node; but instead, permissions are assigned according to the trust degree of nodes and may change dynamically over time based on their behaviorand environment conditions (the context). Permissions are allocated to nodes based on their trust degrees, likewise roles in

RBAC. Such trust degrees are computed based on nodes' behavior in the network. Node behavior is observed directly by nodes or through trusted third parties (TTPs) that are in charge of disseminating recommendations about nodes either among close users or using public key certificates. This solution for access control is distributed and permits each node to be autonomous by establishing by themselves trustthresholds for granting access. However, the solution incurs computation overhead that is not negligible at the node's level; even though, such computations can be delegated to the TTPs distributed in the network to deliver contextual authorizations.

2.2.2 Sensor co-management

SENSEI (Integrating the Physical with the Digital World of the Network of the Future) was a FP7 Integrated Project[44]. The SENSEI created an open, business driven architecture that fundamentally addresses the scalability problems for a large number of globally distributed WSAN devices. In order to realize the vision of Ambient Intelligence in a future network and service environment, heterogeneous wireless sensor and actuator networks (WSAN) have to be integrated into a common framework of global scale and made available to services and applications via universal service interfaces. SENSEI created an open, business driven architecture that fundamentally addresses the scalability problems for a large number of globally distributed WS&A devices. It provided necessary network and information management services to enable reliable and accurate context information retrieval and interaction with the physical environment. The SENSEI project provided a highly scalable architectural framework with corresponding protocol solutions that enable easy plug and play integration of a large number of globally distributed WSAN into a global system – providing support for network and information management, security, privacy and trust and accounting.

The 4WARD project [45] aimedat increasing the competitiveness of the European networking industry and at improving the quality of life for European citizens by creating a family of dependable and interoperable networks providing direct and ubiquitous access to information.These future wireless and wired networks were designed to be readily adaptable to current and future needs, at acceptable cost. The goal of the 4WARD's project was to make the development of networks and networked applications faster and easier, leading to both more advanced and more affordable communication services.The project brought significant innovations for network architectures, In-Network Management and the Network of Information, as well as the use of Virtualization to allow multiple networking architectures to co-exist on the same infrastructure.

The TAS³ Integrated Project [46] (Trusted Architecture for Securely Shared Services) aimedatimproving the European-wide impact on services based upon personal information, which is typically generated over a human lifetime and therefore was collected & stored at distributed locations and used in a multitude of business processes.TAS³ provided a next generation trust and security architecture ready to enable the dynamic user-centric management of policies, and ensured end-to-end secure transmission of personal information and user-controlled attributes between heterogeneous, context dependent and continuously changing systems.

MASTER was a collaborative FP7 project[47], aligned to the strategic objective "Secure, dependable and trusted infrastructures". The project provided methodologies and infrastructure that facilitated monitoring, enforcement, management, and auditing of security compliance, especially where highly dynamic service oriented architectures are used to support business process enactment in single, multi-domain, and iterated contexts.MASTER focused on the regulatory requirements related to IT support of application of security policies to business processes in organizations.

2.2.3 Secure connection sharing

There are some projects that are dealing with interconnecting WSN. P2P approaches enable connecting the networks in an ad-hoc mode. Examples for projects employing P2P are GSN [48], a project that enables interconnection even over 3G mobile networks[49]and ShareSense[50].

2.3 Standards

2.3.1 Privacy Manager

P3P[51], XACML[52] and EPAL[53] specify how user's privacy policies can be set up. This kind of privacy management, though, does not natively support interactions with the user. A user-friendly privacy manager is needed, in order to help the user in the policy definition process.

2.3.2 Sensor co-management

IEEE 802.15.4 standard addresses the security issue in WSNs in three modes: unsecured, Access Control Lists and secured. It uses the AES (Advanced Encryption Standard) procedure for the data encryption but it doesn't support a management of the keys and authentication policies[54].

2.3.3 Secure connection sharing

It is proposed by[38]to bridge the areas of P2P networks for shared WSN networks with the usually for multimedia applications used Universal Plug'n'Play (UPnP) standard[55]. The authors propose to use UPnP gateways in the sensor network to bridge to other WSNs transparently.

The partially standardized IPv6 over low power WPAN (6LoWPAN)standard[40]can be used to give individual nodes access to IP communication. This discards the need for powerful, IP protocol-enabled gateway nodes.

2.4 Commercial products

2.4.1 Privacy Manager

AT&T's Privacy Bird [56] is a P3P [51] compliant browser plug-in that reads P3P policies and displays them in an understandable manner. It allows tailoring warning messages in order to meet user's personal privacy concerns.

2.4.2 Sensor co-management

Some efforts have been made by a couple of companies in the direction of sensor co-management.Motorola[57] proposesthe support for concurrent applications in WSNs based on the Mate virtual machine. Oracle[58] continues to support the Squawk virtual machine on their Sun SPOT system, a small J2ME virtual machine that can execute directly from flash memory and has devices drivers written in Java. Yet, no successful co-managed implementation seems to be on the market.

TinySec[59]– proposes a security architecture for protecting the link layer of WSNs. Its main goals are providing access control,maintaining message integrity, confidentiality of the information, and "replay attack" defense policy.

Also, the ZigBeeAlliance holds a commercial application which implements the IEEE 802.15.4 Standard[60].

2.4.3 Secure connection sharing

For connection sharing, there exist a variety of products. They can be classified into peer-to-peer and client-server approaches.Client-server infrastructures let connections between sensor networks be directed over a central server. They mostly aim at interconnecting sensor networks across the world. Typical projects following this approach are SenseWeb developed by Microsoft Research[61], GSI[64], SensorPlanet[63] (initiated by Nokia) and SensorBase[62].

In P2P architectures, each sensor network acts individually as a peer, and there is no need for a central server infrastructure above the WSAN. The decentralized nature of P2P makes special server products obsolete.

2.5 Summary

Sensor co-management in wireless sensor networks implies the existence of AAA services – namely Authentication, Authorization, and Accounting. This has to be provided in the underlying architecture where both user and device must be authenticated. A significant amount of literature has been written on the subject, and we plan to extend on the existing state-of-the-art towards the realization of secure sensor and actuator co-management mechanisms in conjunction with the requirements of the scenarios which will be considered in the TWISNet project.

The technical side of interconnecting heterogeneous wireless sensor networks is well investigated in literature. There are a number of projects, standards and products that realize practical solutions. The security side on the other side hasn’t received so much attention. On the other hand, connections at the IP level can be secured relatively easy, while policies for ad-hoc connection negotiation will have to be further investigated.

3. ARCHITECTURE

3.1 Overview

This TWISNet architecture is composed of a number of subsystems and modules which operate on the data and control plane. The current section presents the overall architecture of the secure services subsystem and then details each of the modules introduced in this architecture, more specifically, Privacy Manager, Selected Delivery, Connection Sharing and Sensor Co-Management.

3.2 Presentation

The proposed architecture covers a group of sensors, a gateway and an application server as depicted in Figure 1. It is divided into a number of interconnected modules dedicated to certain subtasks of the secure services subsystem, as is presented in Figure 1.Details on each of the modules are described in subsequent sections.

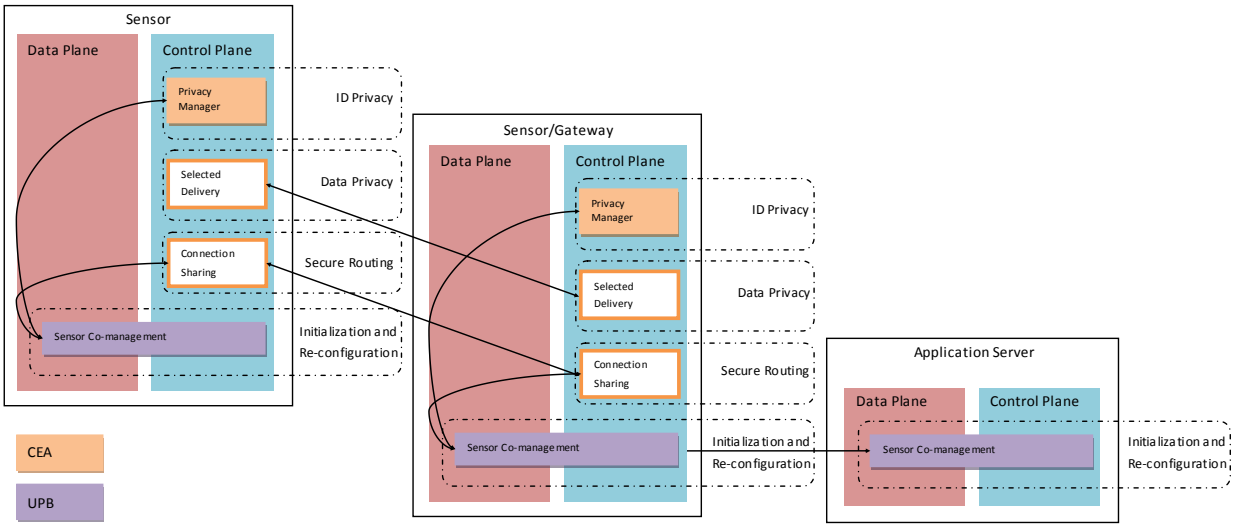


Figure 1: Overview of the architecture for secure services

3.3 Modules Descriptions

3.3.1 Identity Privacy

The need for identity privacy was determined in TWISNet in deliverable D2.1, where the “Sensors Attached to a Person Moving from PAN to PAN” scenario recommended in its threat analysis that appropriate security mechanisms have to be used for securing “the communication channel between PAN device and their data sink”, including an appropriate identity management mechanism.

The Identity Privacy subsystem in task 3.3 is made of a unique module, namely the privacy managerwhich is presented in the following sections. The rationale for this subsystem is that

advanced pseudonymity techniques need advanced tuning, as privacy by itself is not the only requirement to be fulfilled in TWISNet, but has to be provided simultaneously with other requirements, such as energy efficiency or safety. Therefore, there is a need for a central command of privacy mechanisms, able to determine when the use of a given privacy enforcement mechanism has to be triggered and determine which configuration parameters are to be applied in each situation.

3.3.1.1 Privacy Manager Module

The Privacy Manager Module is made of the following components:

- A static database on available pseudonymity mechanisms and their respective configuration parameters, stored along with expected efficiency and energetic (computational) cost;
- A dynamic database of divulged information;
- A decision element. This is the core of the privacy manager, which decides on the pseudonymity scheme to apply at a certain time.

The privacy manager is to interact with the following modules:

- All modules of ID privacy in task 3.2 (that is, enforcement of identity privacy): in order to enforce the determined pseudonymity scheme;
- Security adaptation client module of Security Adaptation subsystem, because the applied pseudonymity scheme may depend on a global security policy, required by a given environmental threat;
- Optionally: Data Aggregation and Processing (DAP) modules (client and/or server) of Secure Data Processing subsystem, in order to determine how the quantitative measurements uploaded by a given sensor may compromise its pseudonymity.

In addition to these interactions, the privacy manager module has to have low-level access to sensor radio outputs, in order to determine how outgoing traffic may compromise node's pseudonymity, depending on outgoing traffic patterns.

3.3.2 Data Privacy

Data privacy is a recommended procedure to protect the information measured or collected by sensors. This subsystem is particularly important in scenario 2 in D2.1: "Sensor Networks for Supply and Demand Optimization" and scenario 3 "Sensor Networks for the Monitoring and Control of an Industrial Process". In both scenarios, sensor data is needed to be accessed in real-time by network devices.

The subsystem comprises two modules that consider access control to data from different perspectives. The selective delivery module, which is described in this deliverable, deals with controlling access to data of a node by another node. On the other hand, the access control module described in D3.6.1 considers access control between the mediation layer and the application layer.

3.3.2.1 Selective Delivery

This module is concerned with accessing network data and is particularly relevant to scenario 3 in D2.1 "sensor networks for the monitoring and control of an industrial process" where field operators might need to access data processed by sensors. With this module, nodes are able to retrieve other nodes' data through a pull or a push model. In the pull model, the node will be authorized to retrieve data from another node relying on the

authorization server (i.e., the trust center). In the push model, the node will push the measured data to a set of nodes based on group communications.

The module is composed of similar counterparts from both sides of the nodes.

The module interacts with the key management module. Keys may be used to prove the possession of the required rights and to access the data. The module will also interact with all modules in the authentication and access control sub-system, and all modules in the ID privacy subsystem, in order to handle dynamic IDs.

3.3.3 Secure Routing

3.3.3.1 Connection Sharing

This module deals with network resources sharing among different domains. For example, the module allows a node to forward its packets through intermediate nodes pertaining to a different domain. The module is more relevant to scenario 4 in D2.1 that is concerned with “multi-owner sensor networks”.

The two counterparts of this module at both nodes are implemented similarly.

The module will interact with the key management module and also all modules in the authentication and access control and ID privacy sub-systems.

3.3.4 Initialization and Re-configuration

3.3.4.1 Sensor Co-management

This module is in charge of secure monitoring of the network on behalf of interested parties, and will come into play in scenarios 2 and 4 of D2.1, “Sensor Networks for Supply and Demand Optimization” and “Multi-Owner Sensor Networks”. Users will be authenticated by the module and given access to sensor data or parameters based on access rights created beforehand. Interaction with SMDR (Secure Monitoring and Device Reconfiguration) will be necessary to relay changes made by users to wireless sensor nodes.

The Sensor Co-management module resides mainly on the Application Server, in the form of a web application that displays sensor data, parameters and allows modification by authorized parties.

4. MODULES TECHNICAL DESCRIPTION

4.1 Privacy Manager

4.1.1 Tasks and Actions

The question of privacy management arises when one considers not only the gains, but also the cost of privacy-ensuring operations. Deliveries of pseudonyms, subsequent auto-configuration operations, setup of corresponding symmetrical resolution systems, are not gratuitous operations, especially in the field of wireless sensor networks. Therefore, there is a need for a trade-off between privacy features offered by a set of privacy-preserving techniques, and the costs that come with these techniques. Optimally adjusting this trade-off to nodes capabilities and requirements is the first reason for a privacy management system. Another rationale for such system is the fact that some privacy preserving techniques, which appear valuable from a local point of view, may actually be useless, if not harmful, when considered globally.

4.1.1.1 Pseudonymity and Unlinkability

Among the privacy services are classically distinguished anonymity, pseudonymity, unlinkability, undetectability and unobservability. TWISNet does not consider the two latter elements of this service list: the fact that a service is being offered (detection) or that a service is being accessed (observation) is not secret per se, as long as the identity of the subject(s) that are accessing it remains obfuscated.

Anonymity is provided at the applicative layer, wherein identifiers can be hidden to the eyes of an unauthorized observer.

Pseudonymity is tightly bound to unlinkability. By using a (pseudo-random) pseudonym for performing an action, a subject ensures that its identity remains hidden. If the pseudonymity model is known to the entity by which the action is performed, a support for accountability is provided, which is generally a requirement for a pseudonymity framework. This is the case in TWISNet, wherein both pseudonymity support and pseudonym resolution are provided; these notions are further elaborated in deliverable D3.2.1. However the static use of a pseudonym is not sufficient to guarantee the anonymity of a node. As the node performs *linkable* actions, the set of potential subjects that it could match restricts itself, which eventually contradicts the definition of node's anonymity quality. Worst case, a node that would keep using the same pseudonym could become unequivocally identifiable through the mere observation of its successive past actionsby an attacker. For example, it may be determined that a mobile node having followed a certain path within a nuclear facility would forcibly have been carried by the only user whose surveillance path includes the same visited areas. Therefore, pseudonymity could not by itself ensure privacy protection: unlinkability is required too, wherein unlinkability is defined as the impossibility for an attacker to distinguish whether two actions are related (that is, pertaining to the same subject) or not.

From the above definition, it results that unlinkability between two distinct actions is only required insofar the linking of these two actions would provide an attacking observer with information allowing it to bypass a privacy scheme in place, typically by allowing it to restrain the set of potential subjects for a certain sequence of linkable actions.

Unlinkability is obtained in TWISNet through the use of pseudonyms. Pseudonyms are meant as mutable identifying elements within a message sent by a subject. By changing its pseudonym between two linkable actions, a subject aims at preventing an observing attacker from linking these actions, thereby compromising its privacy. Of course, this requires that the new pseudonym could not be mapped to the former one. Neither should it be possible for an

attacker to link an action to another through the observation of another element as the pseudonym itself.

Obviously, the decision by a node to change its pseudonym should not happen too seldom; otherwise identifying elements can be retrieved from a set of linkable actions performed by this node. Neither should it happen too frequently, which could lead to a waste of resources. There is therefore a need for a decision mechanism, called hereafter *Privacy Manager*, in charge of this operation. Decision elements taken as input by the Privacy Manager are reviewed in next subsection.

4.1.1.2 Decision Elements

The privacy manager pilots the pseudonymity mechanism enforced by a node in order to achieve privacy protection. It decides when, and optionally how, a node should start using a new (set of) pseudonym(s). This decision is made by the privacy manager based on the following parameters:

- **A set of potential subjects as seen by the observing attacker.** This is a major parameter to be taken into account in order for a node to know when to perform pseudonym change. Since privacy can be defined as preventing an attacker from restraining the set of subjects for a certain observed action, it is important that this set itself be considered in order to induce whether the identification of individual elements part of it will be an easy task for an attacker or not. If one considers TWISNet scenario 1 of a human-carried sensor moving from PAN to PAN, the initial set of potential subjects includes all employees of the facility. This set can be restrained thanks to external attacker's capabilities (as expressed by next two parameters), e.g. if the attacker knows which employees are expected to report data on a given day.
- **The location of the observing attacker(s).** This parameter is related to the current threat model in use in the considered infrastructure. An attacker located in the same wireless network as the observed victim will have access to a different set of information (such as the L2 address, L2 neighbors used for local routing) than an attacker able to monitor the packets received by a sensor node.
- **The capabilities of the observing attacker.** This parameter is closely related to the former. Especially a logical attacker may be physically embodied by multiple physical nodes, or a single attacker may benefit from the assistance of accomplices (e.g. the attacker is eavesdropping packets close to a remote server *and* has compromised sensor nodes in the WSN topology. These collusion attacks are not specifically addressed in the field of TWISNet. Note, however, that the detection of multiple attackers would be an event handled by the threat detection system, which could then issue specific alarms to the network administrator. Besides the location(s) where observation takes place, this parameter also relates to other abilities for the attacker, such as linking ability performed through physical observation in addition to data monitoring. For example, knowing that only a certain employee has moved from a given location to another one at a given time may compromise the privacy scheme put in place by this employee. Finally, this parameter also refers to the ability for an attacker to launch active attacks against a victim, targeting its privacy scheme. Active attacks against privacy are not expected to happen in the TWISNet scenarios, and are therefore not addressed by the TWISNet privacy manager.

- **The linkability of the exchanged information.** The implicit concept that is generally underlying in most privacy schemes is that a data unit sent by a node is composed of a set of identifiers (considered as a global "identifier" for clarity reasons) and an unidentifiable payload. Even though the payload is obfuscated, it would be wrong to believe that only the identifiers leak information that can be used to threaten privacy. This becomes clearer if one extends the notion of identifiability to include linkability:two actions performed by the same subject (especially, two messages sent to or received by the same subject) can be linked to each other even if performed under different pseudonyms. Various pattern analysis criteria may be used by an attacker for doing so. Despite the pseudonymity system it implements, a node could for example be recognized based on :
    - The frequency at which it is sending messages;
    - An identifying pattern within the messages it is sending;
    - The identity of the recipient to which it is sending messages.

  These recognition elements have to be considered globally, among the entire population of potential subjects from the point of view of an attacker. If one considers, for example, that a single node within a given observable infrastructure keeps exchanging data with the same recipient, then obviously all exchanged messages will be linkable with each other, independently of the pseudonymity scheme enforced by the node. Oppositely, if a plurality of nodes is involved in parallel sessions with a common peer, using similar traffic pattern, the attacker will not be able to restrain the potential subjects set from mere traffic observation.

4.1.2 Protocol

The operation of the privacy manager involves observation (gathering of information needed to issue privacy control orders), decision (when and how to set up the pseudonymity enforcement) and action (actual enforcement of the decision).

From the description given in the previous section, it appears that the privacy manager should base its decision on when to update pseudonyms on various parameters, whose observation requires a global view of the potential subjects and their respective ongoing actions. This global view may either be performed on a local (per-node), distributed basis, or by a centralizing entity. Considering the amount of resources required by a sensor in a monitoring operation, it is decided in TWISNet to locate the Privacy Manager on a gateway, which is both more resourceful than a constrained sensor and in a position allowing it to monitor traffic exchanged by a large number of nodes. In such a position, the Privacy Manager is able of serving several nodes, and especially of synchronizing their pseudonymity enforcement. This way, multiple devices can start using new pseudonyms at the same time, which makes the task much more difficult for the attacker to induce that a certain pseudonym, no longer used, has been replaced by a pseudonym that starts being used at the same moment and that both pseudonyms therefore actually refer to a single node.

The decision of the privacy manager to order the use of a new privacy protection scheme is based on a reasoning operation over the parameters described in the first subsection. Once it has been determined that a new privacy protection scheme has to be enforced, a dedicated exchange may take place in order to deliver reconfiguration order (and, optionally, parameters) to the client node. This reconfiguration order may be a new pseudonym or set of pseudonyms, or a seed used to generate these. If a pseudonym is transported, proper confidentiality protection must be afforded to the message transporting it. The fact that an order is being issued may itself be used by an attacker against the privacy scheme using it.

In order to mitigate this, traffic flow confidentiality may be used between the privacy manager and the client nodes that it is serving. However, traffic flow confidentiality forces the client nodes to waste resources for processing non-significant messages, whose only purpose is to deceive a potential attacker; this may not be acceptable for resource-constrained nodes and should be considered only in highest threat levels.

4.2 Operation

This section describes the operation of the privacy manager. Before providing details on the algorithmic implementation, it starts with formalization elements that help characterizing the privacy problem addressed in TWISNet, putting it in perspective with other privacy-specific problems studied in the literature.

4.2.1 Formalization

Most of scientific research works in the field of privacy deal with the protection of data records within data sets, in order to prevent an attacker from gaining access to sensitive information. To that respect, various anonymization schemes are proposed. Accordingly, various attacker strategies are reviewed (e.g. composition attacks, where an attacker has access to multiple data sets with different anonymization patterns [66]). These attack strategies are however difficult to synthesize analytically, since they require a formalization of the attacker knowledge – something difficult to achieve except in specific scenarios ([67], [68]).

4.2.1.1 Criterion

A generic criterion is proposed in [69] to express the resilience of a privacy-sanitized data set. The objective of the attacker being to increase the probability that a target *t* fulfills a certain (sensitive) property *s* while having at his disposal the sanitized data set D^* (obtained from the original dataset D) and his own knowledge K, a sanitized dataset will be characterized by a threshold *c* such that:

$$\max_{t,s} \Pr(t\ has\ s \mid K, D^*) < c$$

Considering the objective of the Privacy Manager to deter linkability, this criterion would translate for a network unlinkability objective to:

$$\max_{t,s} \left(P_i\ same\ origin\ as\ P_j \middle| K, D^*\right) < c$$

with P_i, $P_j$being two distinct packets, D^* being the sequence of all packets as seen by the attacker and K, the attacker knowledge.K is being written as a juxtaposition of two distinct types of knowledge K' and K'' defined such that:

- K' is a static knowledge of the attacker that allows it to link two packets with each other. K' may take the form, for example, of the knowledge that a worker will travel along a given path (in this case K' links two packets with each other by connecting each to a common identifier). Or K' may be based on nodes properties, for example "a sensor node communicates with only one destination at a time".
- K'' is a dynamic knowledge that increases with the number of the past packets seen by the attacker.

4.2.1.2 Network privacy vs. Location privacy

Location privacy has also been widely studied in the literature ([70], [71]). A simple distinction is made in [69] between *offline* anonymity for location traces (where the sanitization of the location data set is performed "as a whole", once the entirety of location-marked data has been collected) and *online* anonymity for location traces (where a privacy enforcing entity obfuscates in real-time private information within location-marked data, while it is being produced). TWISNet network privacy relates to the second, which is made all the more complex as data (a packet) cannot be obfuscated between its generation and its access by the attacker. Obfuscation has therefore to be afforded before the packet is sent, in the form of well-chosen pseudonymity enforcement. This process could be denoted as *implicit online anonymity for network traces*.

Accordingly, another aspect of location privacy for which a denomination is proposed in [69] is that of *sequential releases*: contrary to what happens with the single, one-time publication of an entire data set, location privacy (as well as network privacy) exhibits a dynamic behaviour where data items are published in real-time. This makes an attacker able to build in real-time as well his situation knowledge which corresponds to the K'' dynamic knowledge factor that was introduced above.

4.2.1.3 Adoptability of proposed location privacy measures to network privacy

Among the proposed solutions that aim at providing location privacy, some are very location-specific (e.g. coordinates protective schemes) and cannot be transposed to network privacy. Some others are inefficient, as was already highlighted above in this chapter. The analysis on location privacy conducted in [69] establishes that the unique assignment of a single pseudonymous to a single node cannot satisfy location privacy requirements. Likewise, location privacy cannot be guaranteed on trajectories if all participants update their pseudonymous in a synchronous manner [70].

Nevertheless, two concepts proposed in [69] for location privacy can be directly transposed to network privacy, in order to provide the unlinkability that TWISNet aims to achieve. These are spatial cloaking through *cloaking regions* and the *temporal unlinkability principle*.

- With respect to legacy location privacy, spatial cloaking refers to the obfuscation of geographical data through generalization, this generalization taking here the form of a "vague" localization within a so-called cloaking region. Cloaking regions are defined such that each cloaking region contains at least *k*-1 other users (if *k*-anonymity is to be achieved).

  Applicability to network privacy is not straightforward, as no cloaking is possible for a packet. Nevertheless, network topological granularity may achieve similar results. For example, a group of devices within a common L2 cluster would not be distinguishable by an attacker and could therefore be seen as forming a cloaking region. However, the system could not prevent the number of devices to fall below the desired *k*, in case of node shutdown or mobility.

- Temporal unlinkability principle is defined as follows. If an adversary is able to correctly associate a user id u_j with *m* sequential pseudonyms, $p_j^i, \cdots, p_j^{i+m}$ during times $i, \cdots, i+m$ then this adversary should not be able, under reasonable inference assumptions, to determine, with high confidence, the pseudonym p_j^hcorresponding to u_j at some other point in time $h \notin \{i, \cdots, i+m\}$[69][69].

  Temporal unlinkability relates to the unlinkability objective that the privacy manager reviewed in this chapter aims at achieving. However, here again the above definition

cannot be straightforwardly reused, since not all pseudonymized actions have the same weight with respect to facilitating identification in a network privacy scenario. The underlying rationale stems in the fact that, contrary to the data items collected in a location privacy scenario (typically location/datetime bundles), those in network privacy exhibit variable impact on privacy. This leads us to the weighted event concept, which will be detailed in next subsection.

4.2.2 Privacy Management Process

As explained in 4.1.2, the privacy management process involves the operations of observation, decision and action – which qualifies it as an adaptive process. In addition to these operations and in accordance with the reviewed privacy systems, parameterization describing the desired level of privacy is needed as well. The *parameterization*, *observation* and *decision* functions are described in the following. The *action* step is rather implementation-dependant, and will be focused on in next D4.2 and D4.3 deliverables.

4.2.2.1 Parameter initialization

The privacy management server is initialised with a number *k* playing an equivalent role as in *k*-anonymity systems. An observed packet P_j^t sent by a node *j*, member of a region R, at time *t* can theoretically (that is, "under reasonable inference assumptions") not be identified with a probability greater than 1/*k* as having the same origin as an observed packet $P_{x\in R}^{t-l}$, sent by another node within the same region R at a time (*t-l*) if a privacy enforcement operation occurred between (*t-l*) and *t*. This parameter *k* is therefore relevant to the width of the cloaking region.

Another parameter *m* is related to the tolerance of the temporal unlinkability function. Its use is explained in the decision paragraph.

4.2.2.2 Observation

Through observation, the privacy manager can determine for each node *j* the set of the potential subjects that form *j* cloaking region R_j. It must be noted that although there exists one R_j region for each possible attacker (as seen as a combination of attacker location + attacker capabilities), cloaking regions intervene merely by their populations. In case of multiple individual (non-colluding, as per our hypothesis) attackers, the region composed of the lowest number of devices would be considered.

Contrary to location privacy systems, where the size of the cloaking region is freely configurable through policies, there is not much that can be done when the privacy manager detects that $N(R_j)$ falls below the parameter *k*. A warning could be issued to the (then poorly) pseudonymized node when pseudonymization over such a low population occurs.

In addition to the observation of the cloaking region sizes, the privacy manager has to identify which *events* occur throughout the observed topology. Indeed, not all actions represented by a packet have the same impact on the possible identification by an attacker of the device having sent that packet. The following events and the associated ratings with respect to identification, are proposed in TWISNet:

Event class	Event	Packet observation	Identification weight
Communication	**Communication establishment**	First packet sent by node *j* to a new peer	Medium
	Communication continuation	Packet from node *j* to a peer as part of an ongoing session	Very low
	Session termination	No longer packets sent to a peer	Medium
Mobility	**Mobility (new)**	Node *j* , previously in region R, sends a first packet from region R'	High
	Disappearance	Node *j* , previously in region R, stops sending packets from within region R	Medium
	Staticity	Node *j* sends packets from region R after having already sent packets from that region	Very low

4.2.2.3 Decision

The decision by the privacy manager to trigger the synchronous enforcement of pseudonymity throughout a cloaking region is based on regular computation of a per-node immunity rating IR that is then compared with the threshold *m*, which is one of the two initial configuration parameters. The immunity rating IR for a node *j* is obtained as:

$$IR_n(j) = IR_{n-1}(j) * \frac{N(R_j^n) - 1}{N(R_j^n)} * \frac{1}{w_n}$$

where:

- IR_n is the identification immunity rating at the time of event *n*. IR $\in$ [0; 1] with an IR close to 0 (resp. 1) meaning that the node immunity against identification is weak (resp. strong).
- $N(R_j^n)$ is the size of the population of *j* cloaking region at the time of event *n*.
- $w_n \in$ [0; 1] is the weight of event *n*.

4.2.3 Summary

Figure 2 below schematizes the observation and decision tasks of the privacy manager, as specified in the above subsection.

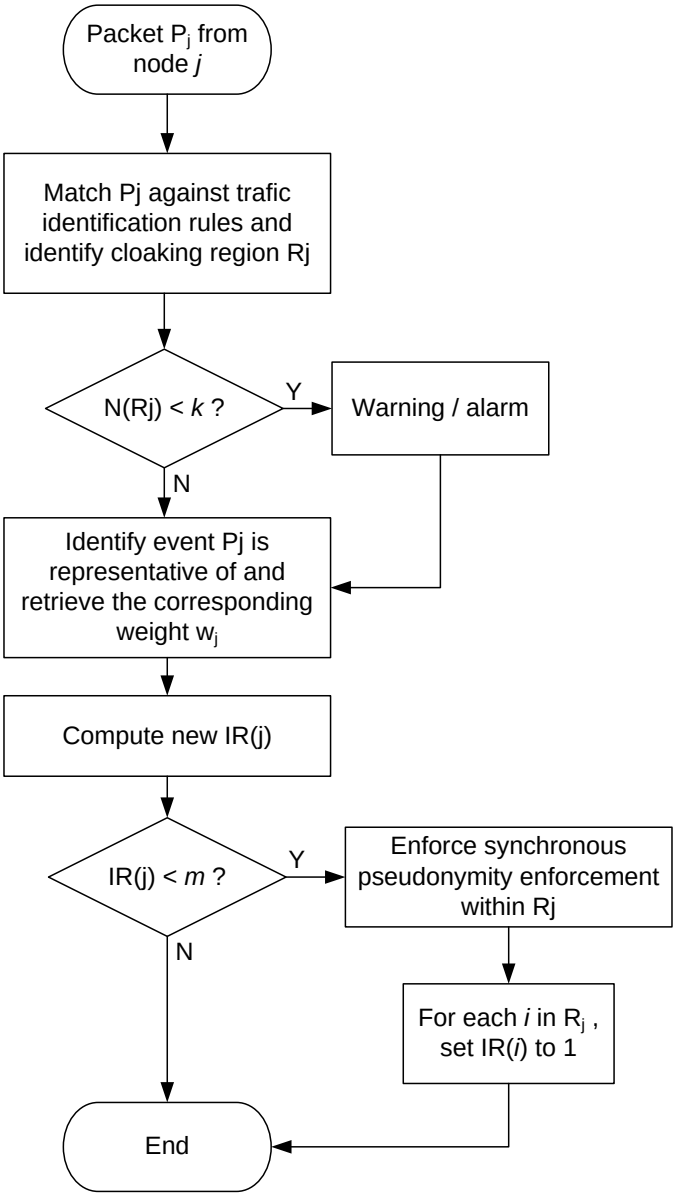


Figure 2: Privacy Manager Operation

4.3 Selective Delivery

4.3.1 Tasks and Actions

The selective data delivery module offers an access control service over sensor network data. Data access control could be carried out through a pull or a push model. In both models, the authorization process is performed by the trust center. Data is delivered by the source node.

The data is delivered in unicast mode to individual sensor nodes or in multicast mode to a group of sensor nodes. In both cases, data is generally confidentiality protected using symmetric keys shared between the data source and the receiver(s).

As proposed in [5], the data access control problem could be reduced to key management problem. To build an access control service over data, a key management system should be put in place. Keys used to access encrypted data are structured into a tree hierarchy in which

high-level keys derive low-level keys. If the node holds a key in the tree, it is able to derive all keys in the sub-tree rooted at the key. This structure is associated with the authorization levels depicting access control hierarchy i.e., nodes with high-level keys can access more sensitive data than nodes with low-level keys.

The mapping between keys and authorization levels is handled by the trust center. This latter is responsible of distributing to nodes the keying materials matching their access rights. Such mapping is managed dynamically by the trust center and supports node join and revocation.

Nodes authorized to gain access to a given data receive the necessary keying material (Traffic Encryption Keys-TEKs) for data decryption, for instancebased on the Multimedia Internet Keying (MIKEY) [65] protocol. Each node is provisioned with a TEK Generation Key (TGK) from which it derives multiple TEKs depending on key position in the key hierarchy structure.

The revocation of rights of a given sensor node requires the update of TEKs at its level and the underlying levels. As a result, it triggers the unicast TGK update at multiple authorized sensor nodes. To limit this problem, TGK update could be grouped for multiple nodes that have a common Key Encryption Key (KEK). Nodes are divided into smaller-groups with a common KEK key that is used to update their TGK key. KEK keys are provisioned to nodes during the distribution of TGK keys. With this approach, TGK update triggers the unicast update of keys at only nodes from the same sub-group as the revoked node. The other nodes receive in multicast the new TGK key encrypted with their common KEKs.

4.3.2 Protocol

In the proposed solution, the keying material received by each authorized sensor node from the trust center consists of TEK Generation Key (TGK) from which Traffic Encryption Keys (TEKs) are derived, along with a set of Key Encryption Keys (KEKs) used to update the TGK. Each TGK is used to derive a set of TEKs structured into a hierarchy of keys similarly to the scheme proposed in [5] for data access control where the high-level keys in the hierarchy generate low-level keys,and therefore permit to have more access rights over provisioned data in the network (refer to Figure 3).

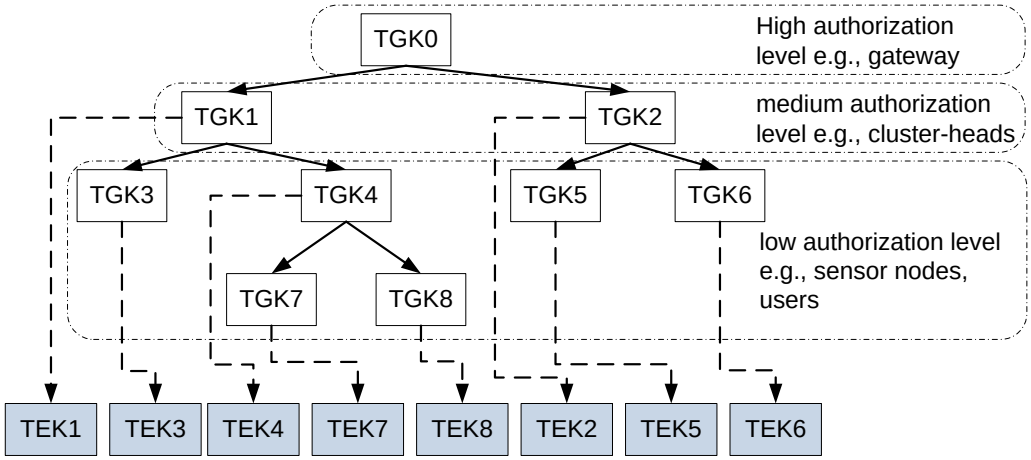


Figure 3: Key hierarchy mapped to node authorization levels

4.3.2.1 Key derivation

Each node is provisioned with a TGK key that corresponds to its authorization level. Lower-level keys are derived based on a cryptographic one-way function f. for example, a low level

key is computed as TGK_low = f(TGK_high, seq), where seqis a sequence number describing the low-level key. To derive a TEK key from TGK, the method used in MIKEY [65] could be applied. After generating TEK keys, the node is able to decrypt the corresponding data streams.

4.3.2.2 Key management

Along TGKs and TEKs, nodes store as well KEKs that are used to update TGKs. Each node will store multiple KEKs shared with some subsets of nodes from the network holding the same TGK (i.e., at the same authorization level as the node). TGK keys stored at nodes are periodically refreshed. KEKs are used to update these keys based on broadcast communications for nodes sharing a common KEK. An access right could be revoked for a node, in this case, all keys (TGK, TEK, KEK) held or could be derived by the node are updated. Depending on the keying materials held by the node, keys are updated:

- *Broadcast TGK update:* for nodes that share a common KEK that is not known to the revoked node, the KEK is used to broadcast new keying materials for these nodes. Obviously, these nodes will pertain to a different authorization level from the revoked node level.
- *Unicast TGK update:* for nodes that have all their group communication keys shared with the revoked node, the trust center provision them with new keying materials based on unicast communication. Their pair-wise keys shared with the trust center will be used to provision them with new keys.

The use of KEKs allows optimizing the communication usage in the network. A large number of KEKs stored at nodes increases the chances that nodes will share KEKs not known to the revoked node and thus broadcast TGK update is employed. However, for a large scale network, nodes are required to store a large number of KEKs which is not practical for resource-limited nodes. The number of KEKs stored at nodes should be chosen taking into account this limitation.
The selective delivery module focuses in providing an efficient approach for controlling access over sensor data. The proposed approach is centered around group key management, and therefore can be integrated into the key management module in D3.1.2, thus allowing the key management module to be capable of managing group keys in an efficient way.

4.4 Connection Sharing

4.4.1 Tasks and Actions

Thanks to the mobility management module applied in a multi-domain context, a mobile sensor node is able to securely connect to a network from a different domain. The connection sharing module allows nodes from a given administrative domain not only to gain access to the network subscribed to a different administrative domain, but also to be accountable for the amount of network traffic produced.

Accountability for connection sharing can be provided using a log system that allows minimal network traffic forensics for performance considerations. Intermediate entities (e.g., gateway, enforcement points) from the foreign network can log minimal information (e.g., number of packets, cumulative traffic size) about the traffic produced by the mobile sensor node.

The log information collected by different network entities is periodically sent to the trust center of the foreign domain network. Based on suchcollected information, the foreign network is able to either limit the traffic produced by the mobile node to a certain threshold or obtain fair fees for network utilization from the administrative domain of the mobile node. If

the first approach is applied, the network revokes network access to the mobile node and notifies its network access enforcement points (e.g., gateway) when the mobile node traffic exceeds the granted threshold.

With the log information, the foreign domain network is able to prove the traffic generated by the mobile node in the foreign network. At the opposite side, the home domain network of the mobile node should be able to verify such log information and optionally assess the quality of service provided by the foreign domain.

4.4.2 Protocol

The connection sharing module should be deployed at network access control nodes (e.g., gateway,enforcement points). These nodes keep track of the traffic generated by mobile nodes from different domains. They send the collected traffic information about these nodes to their domain trust center in a regularbasis to free up their storage space.

4.4.2.1 Log information

The log information about the traffic of a given visiting mobile node includes:

- **The identifier**: the identifier may comprise the identity of the visiting node and/or the session identifier of the exchanged messages. The used identifiers depend on the communication layer at which the module is implemented (e.g., IP address if the module is implemented at the network layer).
- **The numberof packets produced**: this number may consist of the sequence number identifying the last packet sent in the network.
- **The cumulativesize of packets**: this information is relevant if the mobile node produces packets of different sizes. If packets are formatted with identical size, only sequence number can be kept in the log information.
- **A MAC value** or a signature authenticating packet provenance.
- **Other optional information**: identifier of source or destination of packets, time of transmission and reception, round-trip delay time, packet type, etc.

To enable the hosting network to securely log the traffic produced by the visiting nodes, these nodes should add to their packets information enabling the public authentication of the packet provenance. Since the use of public signatures is not practical for resource-constrained devices, a reverse-order hash chain will be used instead. Only the bootstrapping of the hash chain (first use of the chain) relies on a signature-based algorithm. In the same way, the acknowledgement of the receipt of packets will be also publicly authenticated using reverse-order hash chains. Typically, the authentication information included in each packet (or acknowledgement) contains the following elements: {K_i_index, seq, size, MAC(K_i, seq, size)}, where K_i_index is referencing the i^{th} key used from the hash chain (i.e., hash(K_i)=K_i-1), seq is thesequence number of the packet, and size is the cumulative size of the sent packets.

4.4.2.2 Local monitoring of traffic

When receiving a packet from a visiting node, the network access control node checks the included authentication information based on the stored log information (i.e., the previously received hash chain key). If the authentication information included in the packet does not correspond to the actual packet attributes, the packet is dropped. Otherwise, the node updates the log information about the visiting node traffic with this new authentication information.

4.4.2.3 Global monitoring of traffic

The collected information by network nodes is periodically sent to the trust center to free up storage space at these nodes. The trust center is in charge of analyzing the received information and making adequate decisions based on such analysis.

Based on the knowledge of the key in the reverse order hash chain, the trust center can prove that the visiting node has produced an amount of traffic in the network and also that the visited network has correctly forwarded node packets thanks to the authenticated acknowledgements. When analyzing the collected log information, the trust center checks the coherence between the sent packets and their acknowledgements. This check could be performed in advance by network access nodes to prevent visiting nodesfrom cheating.

For both types of monitoring, the connection sharing module can use the functionalities provided by the secure routing and packet forwarding modules describedin D3.2.2. Log information about visiting nodes can be stored and piggybacked as trust and performance metrics introduced by the secure routing module. Decisions on whether or not to forward packets of visiting nodes can be achieved with the packet forwarding module. Therefore, the connection sharing module can be simply integrated into the secure routing subsystem.

4.5 Sensor Co-management

4.5.1 Tasks and Actions

The sensor co-management module (SCM) is responsible for mediation of third party access to parts of the wireless sensor network. Its purpose is to authenticate users coming from the outside of the network and grant them access to sensor data or node configuration. The system will also log user actions in the system. Multiple third parties will have access to the wireless sensor network through this module, each with their own rights within the network.

4.5.2 Protocol

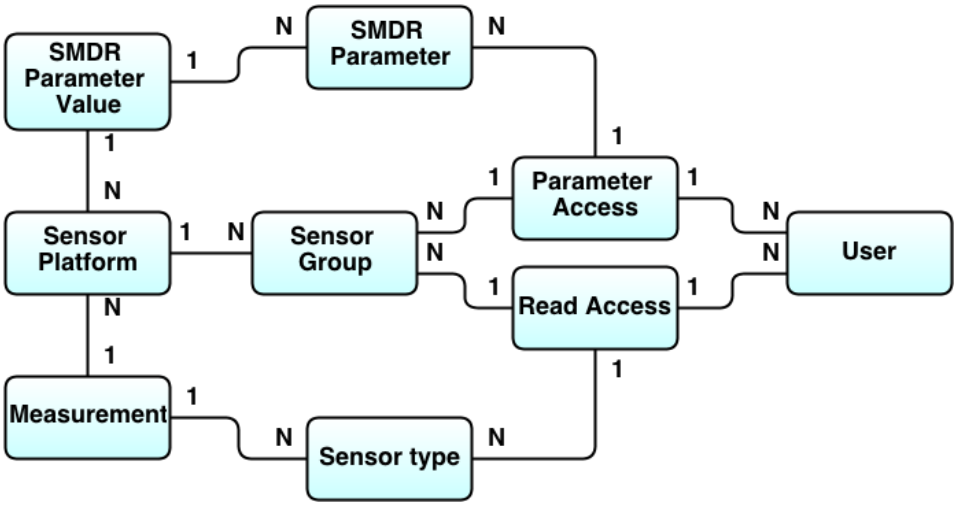
The Sensor Co-Management (SCM) module runs on the Application Server and is tasked with managing all the connected WSNs. It uses a database for storing the access control information and authorizes users to modify parameters or to view data according to the defined restrictions. The database also stores node parameters, as well as application data, and independently receives modifications from server-side components of applications or other modules. The node parameters are updated by the server-side component of the SMDR module according to the information gathered from sensor nodes. Parameter modifications coming from authorized users are written by the SCM module to a special table which is scanned periodically by the SMDR module for propagating all the modifications back to the sensor platforms.

The SCM management interface is accessible using a standard web browser. We use the well known HTTPS protocol to ensure a secure connection with the user's device. This solution offers maximum flexibility for users, who can manage their network from a traditional desktop PC or from a mobile device such as a smartphone or a tablet, a scenario which is very important for home users.

Access to sensor data or parameters is done through access lists, one list for read access to measurement data and a list for node parameter access. The read access list contains items that each give access to a single user to a sensor type on a group of nodes. This means, for instance, that in order to give a user rights to each sensor type available on a group of platforms, entities for granting rights to each individual sensor type present on the platform

must be created. These lists only have to be created once, as the system is deployed, and these steps are automated through the interface available to administrators (privileged users that configure the access lists).

The figure shows the relationship between different entities represented within the SCM module. Each relationship has a number associated on each side, describing how many entities of the other type are linked. For instance, a “Read Access” entity will have 1 User, 1 Sensor group and 1 Sensor Type.



A more detailed description of the entities are as follows:

- User: A logged-in user in the system
- Sensor Platform: A sensor node in the WSN
- Sensor Group: A group of sensor nodes sharing common functionality (e.g. Gas meter sensors, all having the same hardware and owner)
- SMDR Parameter: A parameter on a sensor platform affecting a part of the functionality of the node (measurement rate, transmit power etc.)
- SMDR Parameter Value: A value associated with a known SMDR parameter, on a given sensor platform and at a given time
- Parameter Access: Element in the access list for parameters.
- Read Access: Element in the access list for sensor measurements
- Measurement: A value measured on a sensor type present on a given sensor platform at a given time

5. CONCLUSION

5.1 Summary

This document describes the current status of the work that is taking place in task 3.3 of the TWISNet project. The D3.3.2 deliverable addressed the most important topics that are of interest in the area of security services in multi-owner scenarios. Based on a comprehensive state of the art, four modules have been proposed for integration into the overall TWISNet architecture. These modules are:

- Privacy Manager
- Selected Delivery
- Connection Sharing
- Sensor Co-Management

These four modules offer the necessary functionality for the successful implementation of the scenarios that have been identified and described in the D2.2 deliverable from work package 2.Furthermore, this document proposes a detailed architecture that integrates the four technical modules with precise specifications as to the tasks and actions which are available to those modules, as well as the protocols which implement the desired functionality.

In WP3, the actual implementation of the specifications and the precise interfaces between modulesoffered in this document was completed, and all the interactions of all developed subsystems in the TWISNet architecturewere achieved.

This document contains technical details on the realization of each module that will be implemented in WP4, while the integration in WP4 will be presented in the final demonstrations and deliverables from WP4.

5.2 Outlook

Starting from the industrial scenarios proposed in WP2, we decided on the hardware restrictions of the platform that is used in the project, leading to onlysome ofthe modules being developed in task 3.3and to be integrated in WP4 until the end of the project.

ACKNOWLEDGEMENTS

The TWISNet consortium would like to acknowledge the support of the European Commission partly funding the TWISNet project under Grant Agreement FP7-ICT-STREP-258280.

REFERENCES

[1] Miranda Mowbray , Siani Pearson, “A client-based privacy manager for cloud computing”, Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE, June 16-19, 2009, Dublin, Ireland.

[2] M. Wu,”Adaptive Privacy Management for Distributed Applications”, PhD Thesis, June 2007.

[3] HangyuKo, Seunghyun Kim, Seunghun Jin, “The Real-Time User Consent Acquisition System for Secure Use of Personal Information”, International Conference on Advanced Communication Technology, February 2007.

[4] Oscar Garcia-Morchon , Klaus Wehrle, “Modular context-aware access control for medical sensor networks”, Proceeding of the 15th ACM symposium on Access control models and technologies, June 09-11, 2010, Pittsburgh, Pennsylvania, USA.

[5] A. Sorniotti, R. Molva, L. Gomez, "Efficient access control for wireless sensor data", PIMRC 2008, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Cannes, France, 2008.

[6] M. A. Harrison,W. L. Ruzzo, J. D. Ullman; "Protection in operating systems", Communications of the ACM Vol. 19, 1976.

[7] D. E. Bell and L. J. LaPadula: "Secure computer systems: Mathematical foundations", Technical Report 2547, MITRE Corporation, 1973.

[8] D.F. Ferraiolo and D.R. Kuhn, "Role Based Access Control", National Computer Security Conference 1992.

[9] M. Covington, W. Long, S. Srinivasan, A. Dey, M. Ahamad, and G. D. Abowd, "Securing context-aware applications using environment roles". ACM Symposium on Access Control Models and Technology, ACM, 2001.

[10] IETF RFC 2459 – X509 - http://www.ietf.org/rfc/rfc2459.txt

[11] IETF PKIX http://tools.ietf.org/wg/pkix/

[12] IETF PGP RFC 4880 http://www.ietf.org/rfc/rfc4880.txt

[13] IETF Kerberos RFC 4120 http://www.ietf.org/rfc/rfc4120.txt

[14] Neuman, B.C. Tsaposo, T. “Kerberos: an authentication service for computer networks.” IEEE Communications Magazine. Volume 32, Issue 9, Pages 33 – 38. September 1994.

[15] “Introduction to Windows Live ID”, http://msdn2.microsoft.com/en-us/library/bb288408.aspx (Last Accessed on April 2011)

[16] OpenIDhttp://openid.net/

[17] Shim, S.S.Y., Bhalla, P. “Federated identity management.” IEEE Computer Volume 38, Issue 12, Pages 120 – 122, December 2005.

[18] Liberty Alliance http://www.projectliberty.org/

[19] “Liberty Alliance Identity Architecture Whitepaper”, 2003. download: http://www.projectliberty.org/liberty/files/whitepapers/lap_identity_architecture_whitepaper_final_pdf/ (Last Accessed on April 2011)

[20] Goodner et al.,“Understanding WS-Federation” 2007 download: http://msdn.microsoft.com/en-us/library/bb498017.aspx(Last Accessed on April 2011)

[21] Web Services Specifications http://msdn.microsoft.com/en-us/webservices/aa740689.aspx (Last Accessed on April 2011)

[22] Anderson et al., Web Services Trust Language (WS-Trust), 2005, download: http://specs.xmlsoap.org/ws/2005/02/trust/ws-trust.pdf (Last Accessed on April 2011)
[23] IETF Framework for Admission Control RFC 2753 http://www.ietf.org/rfc/rfc2753.txt
[24] Anderson et al., "XACML Version 1.1: Committee Specification", 2003, download: http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf, (Last Accessed on April 2011)
[25] Lupu, E., Sloman, M., Dulay, N. and Damianou, N., "Ponder: realising enterprise viewpoint concepts", In Proceedings of the Fourth International Enterprise Distributed Object Computing Conference, 2000.
[26] IETF Attribute certificate RFC 3281 http://www.ietf.org/rfc/rfc3281.txt
[27] IETF SPKI RFC 2693 http://www.ietf.org/rfc/rfc2693.txt
[28] IETF Proxy certificates RFC 3820 http://www.ietf.org/rfc/rfc3820.txt
[29] Welch, V., et al., "X.509 Proxy Certificates for Dynamic Delegation", download: http://www.globus.org/alliance/publications/papers/pki04-welch-proxy-cert-final.pdf, (Last Accessed on April 2011)
[30] IETF Policy Tokens RFC 4534 http://www.ietf.org/rfc/rfc4534.txt
[31] IETF Key Note RFC 2704 http://www.ietf.org/rfc/rfc2704.txt
[32] IETF Radius RCF 2865 http://www.ietf.org/rfc/rfc2865.txt
[33] Carrel, D., Grant, L. "TACACS+ Internet draft", 1997, download: http://tools.ietf.org/draft/draft-grant-tacacs/draft-grant-tacacs-02.txt(Last Accessed on April 2011)
[34] IETF Diameter RFC 3588 http://www.ietf.org/rfc/rfc3588.txt
[35] Chappell. "Introducing Windows CardSpace", http://msdn.microsoft.com/en-us/library/aa480189.aspx, (Last Accessed on April 2011)
[36] Digital Me http://www.bandit-project.org/index.php/Digital_Me
[37] OAuthhttp://oauth.net/documentation/getting-started
[38] M. Isomura, T. Riedel, C. Decker, M. Beigl, and H. Horiuchi. "Sharing sensor networks". In *Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on*, page 61. IEEE, 2006.
[39] L. Shu, M. Hauswirth, L. Cheng, J. Ma, V. Reynolds, and L. Zhang. "Sharing worldwide sensor network". In *Applications and the Internet, 2008. SAINT 2008. International Symposium on*, pages 189–192. IEEE, 2008.
[40] IPv6 over Low Power WPAN (6LoWPAN). http://www.ietf.org/html.charters/6lowpan-charter.html.
[41] A. Dunkels, J. Alonso, T. Voigt, H. Ritter, and J. Schiller. "Connecting Wireless Sensornets with tcp/ip networks". In*Wired/Wireless Internet Communications*, pages 583–594, 2004.
[42] UBISEC (Ubiquitous Networks with a Secure Provision of Services, Access, and Content Delivery) project: http://websrv2.c-lab.de/ubisec/
[43] F.A. Mendoza, A.M. López, C. Campo and R.C. García. Trustac: "Trust-based access control for pervasive devices". In: D. Hutter and M. Ullmann, Editors, Lecture Notes in Computer Science vol. 3450, Springer (2005), pp. 225–238.
[44] SENSEI Project : http:// www.sensei-project.eu (Last Accessed on April 2011)
[45] 4WARD Project : http://www.4ward-project.eu (Last Accessed on April 2011)

[46] TAS³ Project: http://vds1628.sivit.org/tas3/ (Last Accessed on April 2011)
[47] MASTER Project: http://www.master-fp7.eu (Last Accessed on April 2011)
[48] K. Aberer, M. Hauswirth, and A. Salehi. “Infrastructure for data processing in large-scale interconnected sensor networks”. In *Mobile Data Management, 2007 International Conference on*, pages 198–205. IEEE, 2008.
[49] S. Krco, D. Cleary, and D. Parker. “P2P mobile sensor networks”. In *System Sciences, 2005. HICSS’05. Proceedings of the 38th Annual Hawaii International Conference on*, page 324c. IEEE, 2005.
[50] A. Antoniou, I. Chatzigiannakis, A. Kinalis, G. Mylonas, S. Nikoletseas, and A. Papageorgiou. “A peer-to-peer environment for monitoring multiple wireless sensor networks”. In *Proceedings of the 2nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, pages 132–135. ACM, 2007.
[51] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, “The Platform for Privacy Preferences 1.0 Specification”, W3C, April 2002.
[52] OASIS “eXtensible Access Markup Language (XACML) Version 2.0”, Committee draft, 2004.
[53] Paul Ashely, Satoshi Hada, Gunter Karjoth, Calvin Powers, Matthias Schunter, “Enterprise Privacy Authorization Language (EPAL 1.2)”, W3C Member Submission, November 2003.
[54] Gascón, David (February 5, 2009). "Security in 802.15.4 and ZigBee networks". http://sensor-networks.org/index.php?page=0903503549. (Retrieved 9 December 2010).
[55] Universal plug and play (upnp). http://www.upnp.org.
[56] AT&T Privacy Bird:http://www.privacybird.org/
[57] Yang Yu, Loren J. Rittle, VartikaBhandar, Jason B. LeBrun „Supporting Concurrent Applications in Wireless Sensor Networks“,SenSys '06 Proceedings of the 4th international conference on Embedded networked sensor systems.
[58] Sun SPOT System using Squawk Java VM http://labs.oracle.com/spotlight/SunSPOTSJune30.pdf (Last Accessed on April 2011)
[59] “TinySec: A Link Layer Security Architecture for Wireless Sensor Networks”. Proceeding SenSys '04 Proceedings of the 2nd international conference on Embedded networked sensor systems, pp. 162—175, ACM, New York, NY, USA, 2004.
[60] Kuorilehto, M., Kohvakka, M., Suhonen, J., Hamalainen, P., Hannikainen, M. Hamalainen, T.D. (2007). “Ultra-Low Energy Wireless Sensor Networks in Practice. Theory, Realization and Development” (pp. 137-138).
[61] Senseweb (microsoft research). http://research.microsoft.com/en-us/projects/senseweb.
[62] Sensorbase project. http://sensorbase.org.
[63] Sensorplanet (nokia). http://www.sensorplanet.org.
[64] C.L. Fok, G.C. Roman, and C. Lu. “Towards a flexible global sensing infrastructure”. *ACM SIGBED Review*, 4(3):1–6, 2007.
[65] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. “MIKEY: Multimedia Internet KEYing”. The Internet Engineering Task Force (IETF) RFC 3830, August 2004.
[66] S. Ganta, S. Kasiviswanathan, A. Smith "Composition attacks and auxiliary information in data privacy," in *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining*, Las Vegas, 2008.

[67] B.-C. Chen, K. LeFevre, and R. Ramakrishnan, "PrivacySkyline: Privacy with multidimensional adversarial knowledge," in *Proceedings of the 33rd International Conference on Very Large Databases (VLDB)*, 2007.

[68] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke and J. Halpern, "Worst-case background knowledge for privacypreserving data publishing", in *Proceedings of the IEEE international conference on data engineering (ICDE)*, Istanbul, 2007.

[69] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala. Privacy-preserving data publishing. Foundationsand Trends in Databases, 2(1-2):1–167, 2009.

[70] Gruteser, Marco, and Baik Hoh. "On the anonymity of periodic location samples." Security in Pervasive Computing. Springer Berlin Heidelberg, 2005. 179-192.

[71] Krumm, John. "Inference attacks on location tracks." Pervasive Computing. Springer Berlin Heidelberg, 2007. 127-143.

[72] L. Sweeney, "K-Anonymity: a model for protecting privacy," IJUFKS 10(5):557–570, 2002.