CHIST-ERA

User empowerment for SEcurity and privacy in Internet of Things

C-ITS and Smart Objects: Working Requirements

Deliverable number: D1.3

Version 1.0

Funded by the Future and Emerging Technologies (FET) CHIST-ERA programme of the European Union.

Project Acronym: USEIT
Project Full Title: User empowerment for SEcurity and privacy in Internet of Things
Call: 2015
Grant Number: 20CH21_167531
Project URL: http://useit.eu.org

Editor:	Gregory Neven, IBM Research – Zurich
Deliverable nature:	Report
Dissemination level:	Public
Delivery Date:	2018-02-28
Authors:	José Luis Hernández Ramos, University of Murcia Gregory Neven, IBM Research – Zurich Alexis Olivereau, CEA Nouha Oualha, CEA Antonio Skarmeta, University of Murcia

Abstract

This document provides an overview of the main use cases and the associated security and privacy requirements to be addressed during the project. In particular, the proposed setting, Cooperative Intelligent Transport Systems (C-ITS) and Smart Objects represent highly timely scenarios where USEIT mechanisms can be leveraged. Rather than creating separate deliverables for both use cases as originally planned, the USEIT project decided to include requirements for both use cases already in Deliverable D1.1. This document further refines the initial requirements described in D1.1. Starting from the experience of previous European efforts, the present document provides a set of initial requirements mainly related to users' empowerment aspects, as well as the challenges associated to the inclusion of cryptographic approaches into constrained devices and networks. These working requirements serve as the baseline to define specific security and privacy mechanisms to realize the proposed use cases.

Contents

List of Figures

List of Tables

Executive Summary

This USEIT deliverable is intended to provide an overview of the main use cases and the associated security and privacy requirements to be addressed during the project. In particular, the proposed setting, Cooperative Intelligent Transport Systems (C-ITS) and Smart Objects represent highly timely scenarios where USEIT mechanisms can be leveraged. Rather than creating separate deliverables for both use cases as originally planned, the USEIT project decided to include requirements for both use cases in all deliverables. This documents contains the working requirements for most of the duration of the project. The initial requirements were published earlier as D1.1; updates to the initial requirements include the tables of numbered requirements on pages 9 and 18, the addition of timing requirements for the C-ITS use case in Section 2.3.4, and more explicit requirements related to smart buildings in Section 3.3. The final requirements will be published as D1.4.

Starting from the experience of previous European efforts, this document provides a set of initial requirements mainly related to users' empowerment aspects, as well as the challenges associated to the inclusion of cryptographic approaches into constrained devices and networks. These initial requirements will be refined in D1.3 and serve as the baseline to define specific security and privacy mechanisms to realize the proposed use cases.

List of Acronyms

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ABE	Attribute-Based Encryption
AES	Advanced Encryption Standard
CAM	Cooperative Awareness Message
C-ITS	Cooperative Intelligent Transport System
CoAP	Constrained Application Protocol
COSE	CBOR Object Signing and Encryption
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
DENM	Decentralized Environmental Notification Message
DTLS	Datagram Transport Layer Security
ECC	Elliptic Curve Cryptography
ETSI	European Telecommunications Standards Institute
HSM	hardware security module
ITS	Intelligent Transport System
LLN	Low Power and Lossy Networks
MTU	Maximum Transmission Unit
OMA NGSI	OMA Next Generation Services Interface
PAP	Policy Administration Point
PDP	Policy Decision Point
PKI	public-key infrastructure
RSU	road-side unit
V2I	Vehicle-to-Infrastructure (i.e., road infrastructure)
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-X (collective term for V2V, V2I, and V2C)
XACML	eXtensible Access Control Markup Language

1 Introduction

In recent years, the concept of smart cities [2] has overwhelmingly emerged as the response to the demographic challenges associated to an increasingly urbanized population [**?**]. Smart cities are intended to bring together different data sources in a city to build a homogeneous ecosystem of services and applications, allowing a sustainable and efficient growth of the city. Making a city smart requires enormous amounts of data to make effective decisions. These data often originate from physical infrastructure, such as traffic lights or streetlamps, or from citizens' devices, such as smartphones or wearables. The data are sent to central data platforms, which can be considered as the brain of the city. According to IBM [**?**], we create 2.5 quintillion bytes of data every day, so much that 90% of the data in the world today has been created in the last two years alone.

Internet of Things (IoT) [3] has become one of the main enablers of smart cities by allowing the exchange of data among heterogeneous devices. As part of the IoT vision, recent advances in wireless communication technologies and ambient intelligence are enabling physical devices with capabilities to capture, process, and communicate data, thereby transforming the "things" into Smart Objects [4] that smart cities will be composed of. The integration of smart objects in our surrounding environment will enable humans to interact with IoT devices, so that they can use their personal devices to leverage the services provided in an IoT-enabled city.

In order to unlock its huge potential and maximize its benefits, however, it is necessary to minimize the risks associated to IoT ipmlementations. In this sense, security and privacy are currently considered as the main barriers for the deployment of IoT on a broad scale [5]. On the one hand, from the technical point of view, such barriers stem from the need to adapt existing security and privacy technologies to be integrated into emerging scenarios. These solutions, mainly designed for web or cloud environments in recent years, need to be tailored to environments where a large number of heterogeneous smart objects will be enabled to exchange data.

Specifically, advanced cryptographic schemes will have to address significant challenges related to the *performance*, *bandwidth*, *flexibility*, and *scalability* requirements of typical IoT environments. Such needs are aggravated especially when interactions involve devices with tight resource constraints, and Low Power and Lossy Networks (LLN). On the other hand, from the social and human point of view, the main barrier is the lack of approaches that allow users to specify their security and (especially) privacy preferences for such interactions. These mechanisms should be designed by considering the involvement of non-technical users who must be able to govern the disclosure of their personal data to other IoT services or devices.

One of the main use cases of smart cities is to make transportation systems safer and more efficient. Smart infrastructure such as connected traffic signs and road sensors can provide valuable information to steer traffic flows and provide real-time information to citizens. Smart vehicles can directly interact with each other and with the smart infrastructure to obtain information on their surroundings as well as updates on traffic situations. These so-called Cooperative Intelligent Transport Systems (C-ITSs) are also a main focus of the USEIT project, to develop better cryptographic mechanisms to balance security, privacy, and safety requirements in Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) interactions.

Another main goal of USEIT is to develop policy-based tools to assist users in governing the behaviour of their smart objects. These tools are intended to be integrated with advanced cryptographic schemes, in order to minimize the gap between technical and social security and privacy challenges in USEIT scenarios.

2 C-ITS Initial Scenario and Requirements

The next generation of vehicles will be able to communicate with other vehicles, with road infrastructure, and with traffic-monitoring central stations via several communication channels. The main goal is to improve safety and efficiency on the roads, but third-party services may also make use of this infrastructure, e.g., for toll roads or parking. This communication obviously needs to be protected, mainly to prevent attacks that could create safety hazards or disrupt traffic, but also against eavesdropping which could be used to track individuals or to intercept confidential communication.

The security requirements for C-ITS have been studied in several European research projects, including EVITA [6], SeVeCom [7], simTD [8], PRECIOSA [9], and PRESERVE project [10], as well as by standardization bodies, e.g., the European Telecommunications Standards Institute (ETSI) [11]. We summarize here the main requirements from the PRESERVE project [10] and ETSI [11]; the former is itself a summary of the requirements of all sources mentioned above, as well as a paper by Henniger et al. [12].

2.1 C-ITS Communication Setting

2.1.1 Participants

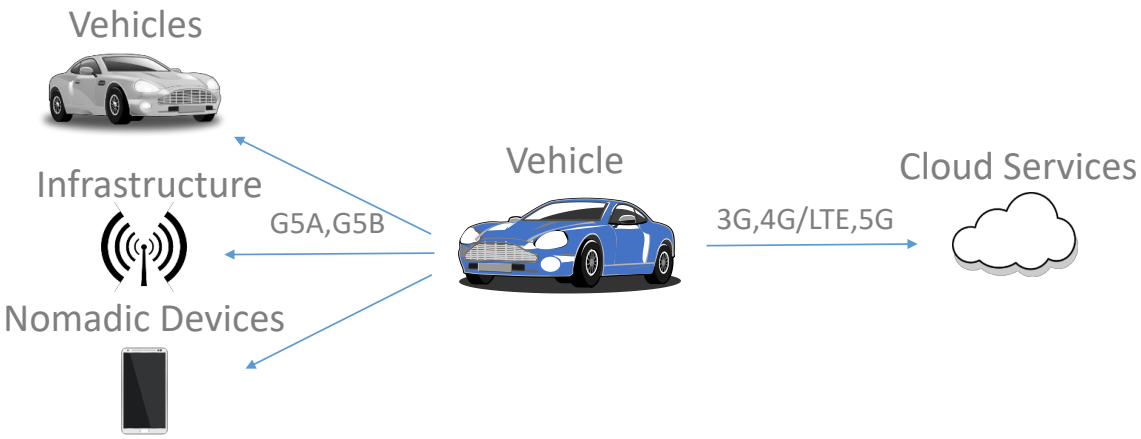

Figure 2.1: V2X communication participants

The different communicating participants in an Intelligent Transport System (ITS) are depicted in Figure 2.1. The main participants are obviously vehicles, which include private vehicles, commercial vehicles (e.g., taxis, buses, or trucks), and priority vehicles (e.g., ambulances). A second category are road-side units (RSUs), which can be sensors, traffic signs, or communication infrastructure. A third type of participants are back-end systems, which are Internet-connected services that can be relied upon for internal services such as public-key infrastructure (PKI) or traffic management, or for third-party services such as weather information, toll collection, or parking. Finally, there are nomadic devices, which can be used by other traffic participants such as bicyclists and pedestrians to participate in ITSs. These may be dedicated devices, or personal electronic devices (e.g., smartphones) that double as ITS devices.

2.1.2 Communication Channels

V2V and V2I communication, jointly referred to as Vehicle-to-X (V2X) communication, mainly takes place over dedicated IEEE 802.11p radio frequency bands ITS G5A and ITS G5B. The former, between 5.875 GHz and 5.905 GHz, is reserved for ITS road safety communication, while the latter, between 5.855 GHz and 5.875 GHz, is reserved for non-safety-related communication. A RSU can also use G5A to communicate with other RSUs, but it may also be connected through a dedicated wired or wireless link. Due to the high mobility of ITS vehicle stations, these communication channels can be rather unstable and suffer from frequent communication disruptions. The transmitters typically have a range of about 300m, with a transmission delay of 3-8ms. Total bandwidth on these channels is about 6 Mbps in total, allowing a peak capacity of about 2200 packets of 200 bytes per second, but practical maximum rates will be rather around 1000 packets per second. With very high traffic density, however, packet loss will increase, causing throughput to decrease.

ITS G5C is commercial WLAN (IEEE 802.11a/b/g/n) that can be used in ad-hoc mode for broadcast or unicast of non-safety-critical functions. Vehicles, RSUs, or traffic-monitoring central stations may be equipped with mobile communication units to transmit information and connect to the Internet over commercial mobile channels (3G, 4G/LTE, 5G). It offers much higher data volumes than ITS G5, but has higher latency too. Connectivity could be unreliable, however, especially in tunnels, parking garages, or remote areas.

Another means of communication for vehicles is via wired or wireless on-board diagnosis tools, as is primarily used in workshops for software updates, diagnosis, or regular maintenance. Finally, electrically-powered cars may be able to use a powerline connection via the charging cable at charging stations.

2.1.3 Types of Messages

Vehicles send two main types of messages over ITS G5A and ITS G5B channels:

- Cooperative Awareness Messages (CAMs): These are beacon messages that vehicles send at regular time intervals, typically at a frequency of 1-10 Hz, containing the vehicle's position, speed, heading, measurements, sensor accuracy, as well as a timestamp. The packet will also contain a confidence measure for all sensor data. The receiving ITS stations use these messages to construct a "local dynamic map" of all surrounding vehicles for various safety-related purposes, e.g., early collision warnings or emergency vehicles warnings. CAMs are not meant to be forwarded in general, but roadside units may forward them, e.g., at intersections with bad network coverage.
- Decentralized Environmental Notification Messages (DENMs): These messages contain notifications about specific safety-related events and hazards, e.g., electronic brake lights, wrong-way drivers, traffic jams, unexpected stationary vehicles, or dangerous road conditions (slippery roads, bad visibility, precipitation). They typically contain a description of the event type, together with severity, location, target region, detection time, and duration of the event. A DENM can be used to signal the detection of an event, an update to a previously detected event, or the revocation of an event. Unlike CAMs, DENMs are often forwarded to more distant nodes.

Other messages could include local service announcements, e.g., to support in-vehicle display of road signs, and speed limits, to notify the driver about nearby points of interest. Other uses include enhanced route guidance, optimal speed advisory, toll collection, vignette requirements, or access control to buildings and parking lots.

2.2 Main Use Cases

The following Basic Set of Applications is a list of use cases and services compiled by ETSI [11] that are mandatory to be deployed in an ITS station.

- A stationary vehicle at a potentially dangerous location, e.g., after an accident or a mechanical problem, sends out warnings to other vehicles. The information may be forwarded by other vehicles, RSUs, or to a traffic management centre.
- Vehicles that detect potentially hazardous local traffic conditions send warnings to other vehicles. Such conditions can include traffic jams ahead, dangerous road conditions (precipitation, road adhesion, visibility, or wind).
- Vehicles and RSUs broadcast warnings when they detect a collision risk or signal violation is detected (e.g., wrong-way driving, traffic light or stop sign violation).
- A mobile RSU distributes messages to warn approaching vehicles about upcoming road works.
- Traffic signs broadcast information to be displayed in the vehicle, including regulatory or contextual (i.e., variable) speed limits.
- RSUs broadcast traffic information and recommended itineraries, e.g., in case of blocked roads, limited access, or detours.
- Emergency vehicles periodically broadcast their position, speed, and heading, as well as whether it has its siren or blue light in use.
- Slow vehicles periodically broadcast their position, encouraging other vehicles to overtake.
- Motorcycles broadcast their presence, just as any other ITS station, but with a special indication that this is a special type of vehicle.
- Vehicles broadcast warnings in case of a sudden braking maneuver or slowdown, thereby functioning as electronic emergency brake lights.
- Traffic lights broadcast timing data reflecting its current state, the time remaining before switching to the next state, and an optimal speed advisory to cross subsequent traffic lights.
- RSUs periodically send information about local services, e..g, financial or insurance services, or fleet management services. Vehicles can establish a unicast connection to request more information.
- Access-controlled areas or parking lots are protected by means of a RSU that broadcasts the access control restrictions. Vehicles entitled to access the area send their credentials through a unicast connection to the RSU.
- Vehicle software/data provisioning and update A road side unit periodically broadcasts the presence of vehicle software access. Vehicles establish a unicast connection and uses the services.
- RSUs with internet access capabilities could provide internet connection to passing vehicles, e.g., to synchronize information between the vehicle and the drivers' or passengers' remote home system.

The actions that a receiving ITS station can undertake upon receiving a relevant message can vary widely, e.g.,

- notifying the driver or passengers by playing a sound or displaying a message
- sending messages to other parties,
- activating certain vehicular components, which could be driving-related components (e.g., turn indicators) as well as non-driving related ones (e.g., air conditioning, changing the entertainment system volume, (de)activating voice communications, or (un)locking doors),
- reconfiguring the vehicle to reduce crash impact, e.g., deploy airbags,
- intervene directly in the driving by actuating the brakes, steering, or accelerating.

2.3 Requirements

We focus mainly on the security and performance requirements that a cryptographic solution for C-ITS has to satisfy. We also give a performance and cost requirements.

2.3.1 Security Requirements

2.3.1.1 Attackers

Any cryptographic solution must envisage external as well as internal attackers. External attackers are outsiders that gain access to the C-ITS communication by placing a reception or transmission device within range of the ITS G5A signals, by obtaining physical access to an ITS station, or by hacking into one through a wireless interface or the Internet.

Internal attackers have full access to one or more ITS vehicle stations, roadside units, or backend systems. Full access also includes access to cryptographic keys stored on the ITS station or backend system, even if such access may be hardened through hardware or software protection (e.g., hardware security modules (HSMs) or secure execution environments). One could also envisage semi-internal attackers who have access to all information on regular storage and who can interact with the hardware of software-secured modules, but do not have access to the secrets in secure storage themselves.

Attackers can be passive, who try to glean information from eavesdropping on the communication, or active, who can actively intervene in the system.

Attackers may have various incentives to attack a C-ITS system, including:

- causing physical harm to drivers or passengers in criminal acts or terrorism,
- vandalism, causing damage to vehicles or infrastructure,
- disrupt traffic or harm the economy of ITS and road traffic, e.g. by provoking traffic congestions,
- denial of use of the road infrastructure,
- manipulation of traffic conditions to the attackers' advantage. The attackers could be local residents in a neighborhood or drivers participating in traffic. The motiviation for this attack range from more comfortable living conditions to a faster commute or even to clear an escape route for criminal purposes.
- gaining reputation as a scientist or hacker,
- identity theft, impersonation of a victim to obtain financial or other advantages,
- obtaining information to build location or driving behavior profiles of the driver or passengers. Individuals may do so to spy on spouses in divorce cases, steal or hijack cars, burglar homes, blackmail drivers when they visit compromising locations, or stalk or abduct people. Companies may do so for financial gain, e.g., to calculate individual driving risk factors or to improve directed advertising. Government bodies may do so as part of unauthorized mass surveillance.
- insurance fraud, by manipulating data that may be stored in vehicles after an accident,
- infringement on the intellectual property of vehicle manufacturers,
- manipulation of vehicles of competing manufacturers to blackmail them or tarnish their public reputation,
- tuning car software.

2.3.1.2 Technical Security Threats

To be more concrete from a technical perspective, we distill the following technical threats that a C-ITS security architecture must protect against:

- Denial of service: making the system unavailable, e.g., through jamming or flooding the network with high volumes of messages. It is hard to completely protect against such attacks, but they should not be further facilitated, e.g., as a consequence of using multi-hop broadcast messaging.
- Malware: introducing malicious code into vehicles or infrastructure.
- Spoofing, impersonation, forgery: impersonating a vehicle with different properties than the actual vehicle, e.g., to obtain the status of an emergency vehicle.

- Spoofing sensor data.
- Replay of previously broadcast messages, either at the same location as where the message was originally sent, or at a different location.
- Compromise key material through side-channel attacks or physical access to ITS stations.
- Sybil attacks, by which a single attacker can impersonate many different vehicles simultaneously, e.g., to emulate a traffic jam. The number of different vehicles that an attacker can emulate should be proportional to the number of ITS stations that the attacker must compromise and extract cryptographic keys from.
- Illegitimate access or manipulation of backend databases.
- Privacy: tracking of vehicle locations, destinations, or driving patterns.
- Identification: linking of C-ITS messages to the identity of the driver or passengers.

2.3.1.3 Concrete Security Requirements

The first category of security requirements are related to authenticity of senders and claimed attributes. It should not be possible for an external attacker to pose as an ITS station. An internal attacker that compromised one or more ITS stations should not be able to impersonate any honest ITS stations, e.g., to circumvent access control restrictions, or to deceive users and authorities into believing that some uncompromised ITS station was responsible for sending certain messages. The attacker should not be able to claim attributes that are not possessed by any of the compromised ITS stations. Information sent to or from ITS stations should be protected against unauthorized or malicious modification or manipulation during transmission, and it should be impossible to make ITS stations accept packets that were replayed from a different time or location.

A receiving ITS station must be able to verify attributes of the sender and reject incoming messages from unauthorized sources. An unauthorized user must not be able to make an ITS station accept and process any management or configuration information. The ITS stations themselves, and in particular, the cryptographic and management information they hold, must be properly protected against unauthorized modification, and deletion.

The second category of security requirements are related to privacy. It must be impossible for an unauthorized party, even an internal attacker who compromised any number of ITS stations and at most one ITS authority, to link transmitted messages to the true identity of either the user or the user's vehicle by analyzing communications traffic flows to and from the vehicle. It must also be difficult to link, to track the location, to track the route taken, or to establish driving behavior profiles of individual users or vehicles over longer durations of time (maximum 5-10 minutes). Messages should therefore not include a persistent user identity. If some form of identifiers is used, they must be unlinkable and the ITS station must have a means to obtain or derive as many pseudonyms as it needs for the lifetime of the vehicle.

The third category of accountability and revocability is at least as important as, but often in conflict with the privacy requirements. In the event that an ITS station is detected to be providing misleading information to other vehicles (either by malfunction or malicious intent), it must be possible to prevent other units from accepting and processing its messages. Misbehaving messages could be detected, e.g., by performing plausibility tests on incoming messages and reporting suspect packages. The privacy properties of C-ITS messages of course make it more difficult to identify and remove a misbehaving ITS station. Nevertheless, it should be possible to remotely shut down a misbehaving ITS station if it is causing problems to other ITS users, while maintaining the ability to receive messages from other stations. It should also be possible to deactivate special equipment designed specifically for an attack on the ITS system.

Finally, in spite of the fact that CAM and DENM messages are typically broadcast messages, there are also a number of confidentiality requirements. In particular, non-broadcast information sent to or from an authorized ITS station (e.g., management information or unicast messages) should not be revealed to any party not

authorized to receive the information. Information held within the ITS station, in particular, cryptographic key material, should be protected from unauthorized access.

2.3.2 Performance Requirements

An ITS station must be able to securely handle an incoming packet rate of about 1000-1600 packets per second and an outgoing packet rate of about 15 packets per second. The packet delay must be limited to 50 ms. This time includes the time to prepare the outgoing packet on the sender's side, the transmission time, as well as the processing and verification of the packet at the receiver's side.

2.3.3 Non-Functional Requirements

The solution must be economically feasible, meaning that the added cost of hardware and software per vehicle must be limited to 50–100 Euro, and that the operating cost for a fleet of a million vehicles must be limited to 10 Euro per vehicle per year. The solutions must be compatible with interational standards and have minimal legal restrictions that could hinder the deployment, such as cryptography export rules and intellectual property rights.

2.3.4 Timing Requirements

For safety-related use cases such as collision avoidance or road hazard warnings, time is critical. C-ITS messages must be sent out with a latency and frequency that enables vehicles to steer clear from obstacles or come to a complete stop to avoid danger. This means that warning messages must be sent, transmitted, and processed in time for the vehicle to initiate an emergency braking maneuver and come to a complete stop before hitting the obstacle.

The total stopping distance of a vehicle is composed of the braking distance, i.e., the distance that the vehicle travels from the moment its brakes are applied until it comes to a complete stop, and the reaction distance, i.e., the distance traveled between the warning message being displayed and the human driver applying the brakes. A standard perception-reaction time is 1.5 seconds [13], so the reaction distance

The most time-critical case is that of a head-on collision between two vehicles traveling at full speed, e.g., on a highway. Even in such a situation, each vehicle must receive at least one CAM message from the other vehicle that allows both drivers to bring their vehicles to a complete stop. We define the following variables and typical values [13] for our calculation: In the worst case, both vehicles send a CAM message slightly

$v = 130$ km/h $= 36.11$ m/s	speed of the vehicles
$H = 500$ m	transmission horizon, i.e., the distance within which CAM messages can be received by a standard C-ITS receiver
f	broadcasting frequency in Hz, i.e., rate at which CAM messages are sent out
$l_t \approx 0$	transmission latency, i.e., the time that the CAM message travels from one vehicle to the other, negligible for direct radio
$l_p = 50$ ms	processing time, i.e., the time from the receipt of the CAM message to the display of a warning message
$T_{pr} = 1.5$ s	perception-reaction time, i.e., the time between a warning being displayed and a human driver applying the brakes
$\mu = 0.7$	kinetic friction coefficient
$g = 9.81$ N/m^2	gravitational constant
$BD = v^2/(2\mu g) = v^2/13.73$	breaking distance

beyond each other's transmission horizon, so that the next CAM messages are sent only $1/f$ seconds after the vehicles entered each other's transmission horizon. For both vehicles traveling at speed v to be able to come to a complete stop, we need that

$$\begin{aligned} H &\geq 2 \cdot v \cdot (1/f + l_t + l_p + T_{pr}) + 2 \cdot BD \\ &\approx 2v \cdot (1/f + 1.5) + v^2/6.9 \end{aligned}$$

from which one can easily derive the minimal broadcasting frequency at any speed v as

$$f \geq \left(\frac{H}{2v} - \frac{v}{13.8} - 1.5 \right)^{-1} .$$

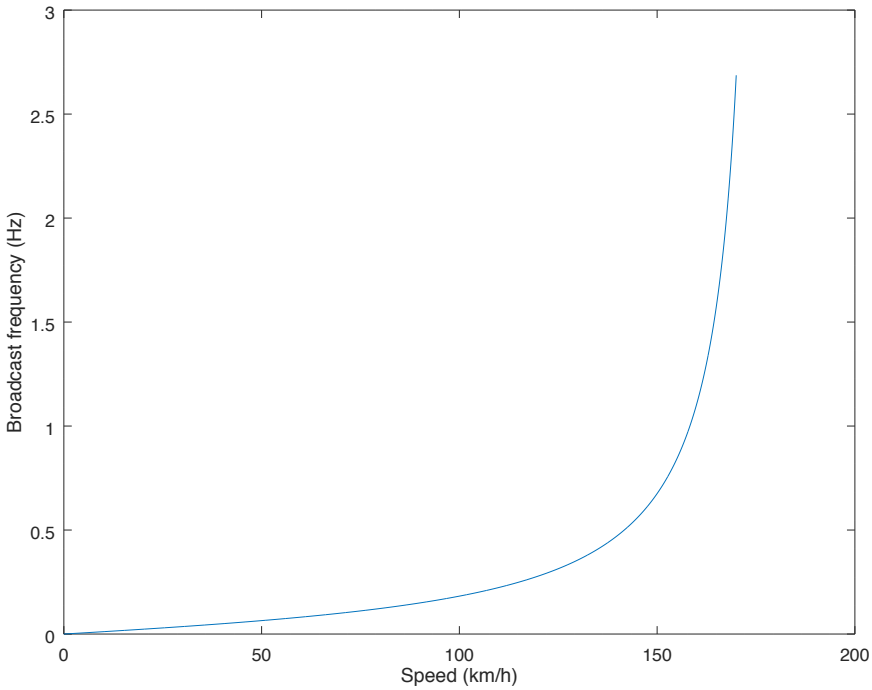

Figure 2.2: Minimal broadcasting frequency of CAM messages in order to avoid a head-on collision between human drivers as a function of the speed of both vehicles.

The resulting minimal broadcasting frequency for different vehicle speeds is shown in Figure 2.2. For speeds above 175 km/h, there is no broadcasting frequency that will allow vehicles to stop in time, because the total stopping distance is greater than the transmission horizon. Note, however, that this is for a situation where *both* vehicles drive at 175 km/h, meaning they are approaching each other at a relative speed of 350 km/h. For two vehicles speeding head-on at a more reasonable highway speed of $v = 130\ \mathrm{km/h} = 36.11\ \mathrm{m/s}$, we obtain a minimal broadcasting frequency of only $f = 0.35$ Hz, or about one message per three seconds, which is considerably less than the previously recommended frequency of 10 messages per second [10]. For vehicles driving at $v = 50\ \mathrm{km/h} = 13.89\ \mathrm{m/s}$, one can even lower the broadcasting frequency to 0.065 Hz, or one message every 15 seconds. The figure clearly shows that for the use case of avoiding head-on collision, which should be one of the most demanding ones, there is no reason to broadcast at frequencies above 1 Hz.

2.3.5 Summary of Requirements

We summarize the requirements for the C-ITS use cases in the table below.

ID	Type	Priority	Description	Rationale	Dependencies
CITSREQ01	SEC	MUST	Protect against impersonation	External attackers cannot pose as ITS station. Internal attackers cannot pose as other ITS stations or claim attributes they don't have.	-
CITSREQ02	SEC	MUST	Protect privacy	Internal attackers cannot link message to user's or vehicle's identity. Difficult to link messages by same vehicle over longer amounts of time. Difficult to create individual user profiles with location, routes, or driving behavior.	-
CITSREQ03	FUN	MUST	Support revocation	Misbehaving ITS stations must be excluded to prevent other from accepting its messages.	-
CITSREQ04	SEC	MUST	Protect against Sybil attacks	An attacker that compromises one ITS station should not be able to emulate many vehicles simultaneously.	-
CITSREQ05	SEC	MUST	Protect integrity	Impossible to modify packets during transmission.	-
CITSREQ06	SEC	MUST	Protect against replay attacks	Impossible to replay packets in a different context (time or location).	-
CITSREQ07	SEC	MUST	Protect against unauthorized access	Unauthorized user should not be able to modify management/configuration of ITS stations or extract cryptographic keys.	-
CITSREQ08	SEC	MUST	Protect confidentiality of restricted content	Prevent unauthorized access to content of restricted messages (management, unicast).	-
CITSREQ09	PFM	SHOULD	Computational efficiency	Handle incoming packet rate of 1600 packets/s and outgoing of 15 packets/s, delay of 50 ms.	-
CITSREQ10	PFM	SHOULD	Signature size	Signature fits in 200-byte packets.	-
CITSREQ11	PFM	SHOULD	Economically feasible	Added cost per vehicle at most 50-100 Euro, operating cost limited to 10 EUR per vehicle per year.	-

3 Smart Objects initial Scenario and Requirements

The main purpose of the Smart Objects scenario is to make emphasis of some of the major security/privacy requirements by abstracting many of smart city use cases, in which information is shared through typical central data management platforms. For this, we use the definition from [4] to define a smart object as any *"autonomous physical/digital objects augmented with sensing, processing, and network capabilities"*. Furthermore, we adopt a *producer/consumer* approach, in which a smart object can act as a data producer or consumer at any time throughout its lifecycle, independently of the communication pattern (e.g. client/server) being employed. In the scope of USEIT, an instantiation of this general setting is considered in a *smart building*, in which smart objects are *things* physically deployed in a building producing data that are consumed by other smart objects or services.

3.1 Smart Objects Communication Setting

The Smart Objects scenario is intended to represent different use cases where smart objects' data are shared across central data platforms to be processed, and shared with other devices, users or applications. Particularly, in such scenarios, physically deployed smart objects (acting as a producer) sense and process data about its surrounding environment. Depending on the specific use case, these devices could be characterized by having strong resource constraints, and operating through LLNs (e.g. based on IPv6 over Low power Wireless Personal Area Networks (6LoWPAN)), which hinders the application of traditional security mechanisms or advanced cryptographic schemes. These devices are usually connected to the Internet through more powerful devices to assist constrained nodes for complex tasks. Smart Objects' data are shared with groups of users or services (acting as a consumer) through data management platforms serving as the *brain* in a particular scenario. This component is additionally intended to apply data mining techniques to extract knowledge from the information that is transmitted from data sources. In such a setting, the main challenge is to maintain typical security and privacy properties in scenarios where the involved producer(s) and consumer(s) entities are decoupled or even they do not know each other.

3.1.1 Participants

Taking into account the considered setting, Figure 3.1 shows an overview of the general Smart Objects scenario. In this sense, the main entities to be considered are:

- Smart Object. They are physical devices acting as data producers and consumers. Smart objects embrace heterogeneous entities including battery-powered and resource-constrained devices, as well as legacy devices (i.e. without IP connectivity).
- Gateway. It provides the ability to interconnect smart objects by abstracting their physical details, and providing a common model to represent and send their data to other smart objects, users, or services. Depending on the smart object's nature, a gateway is intended to assist smart objects to perform their tasks (e.g. to connect with other entities or to perform resource-demanding cryptographic operations).
- Platform. It is responsible for managing all the information obtained from smart objects (through gateways) and disseminating to high-level services. The functionality of the platform is twofold: on the one hand, it serves as an information store allowing communicating entities to remain uncoupled. On the other hand, it is used to perform data analysis tasks in order to provide high-level services based on knowledge extraction.

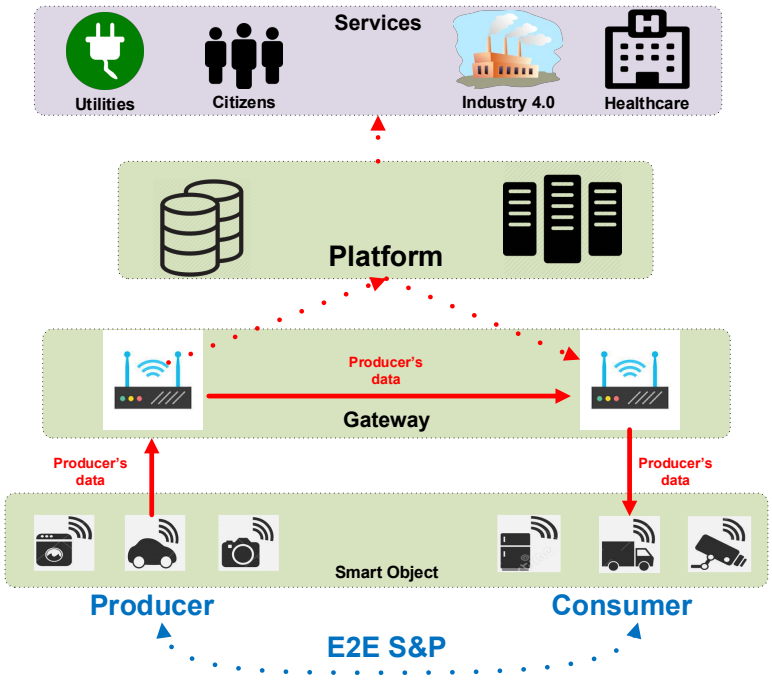

Figure 3.1: Smart Objects scenario overview

- Services. Usually, they represent different applications in a city that are interested on the information provided by the platform in order to provide different utilities to citizens and other services.

Under this general perspective, USEIT will address security and privacy requirements by following two main complementary approaches. On the one hand, it will define concrete platform components for users' empowerment, so that they can define their privacy and access control preferences on their devices' data disclosure. On the other hand, it will analyze the integration of advanced cryptographic schemes into smart objects with strong resource constraints, in order to ensure end-to-end security and privacy properties.

3.1.2 Communication Channels

Given the heterogeneous nature of smart objects and networks in IoT, many use cases are based on the use of different communication protocols. Indeed, protocols widely used on the Web are still used on IoT scenarios in domains where bandwidth and network performance are not a problem (i.e. gateway-platform communications). However, in interactions where resource-constrained smart objects are involved, those protocols have been adapted to these environments. Figure 3.2 summarizes the transition of Web protocols to the protocol stack that is commonly used in the IoT domain.

According to Figure 3.1, Gateway-Platform communications are typically carried out using the *Web Protocol Stack*. This way, current deployments are still valid for interacting with such components and services. In this case, the challenge is not related to the application of security protocols and cryptographic schemes, but to the need for defining scalable mechanisms that allow the security management and abstraction of large numbers of smart objects and associated data.

In the case of smart object-gateway interactions (or even among smart objects), the *IoT Protocol Stack* is commonly used. In this case, Constrained Application Protocol (CoAP) [14] is employed as a lightweight alternative to the use of HTTP. In addition, CoAP defines a security binding to Datagram Transport Layer Security (DTLS) through the use of pre-shared keys, raw public keys or certificates. Then, datagrams are sent through 6LoWPAN, which represents the adaptation of the IPv6 protocol to be employed on constrained environments such as IEEE 802.15.4 networks, in order to obtain end-to-end connectivity between constrained

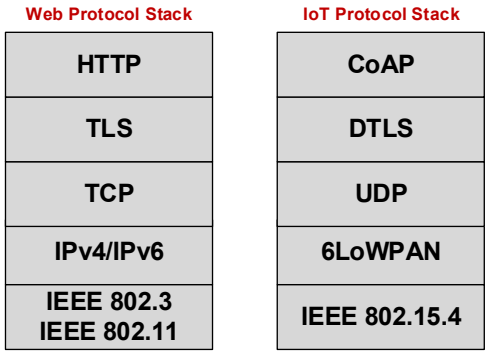

Figure 3.2: Current Web vs IoT protocol stack

devices and any entity connected to the Internet. In this context, the Maximum Transmission Unit (MTU) is 127 bytes, so any larger packet needs to be fragmented with consequent network degradation. These limitations have a direct impact on the application of cryptographic schemes. While the use of symmetric cryptographic schemes (such as Advanced Encryption Standard (AES)) do not represent an issue in terms of bandwidth, their scalability regarding key management and revocation aspects with scenarios with a large amount of smart objects, represents a clear limitation. On the contrary, an approach based on public-key cryptography can be challenging due to the size of keys, ciphertexts and signatures. This problem is exacerbated in the case of the use of RSA-based schemes, and consequently, alternative Elliptic Curve Cryptography (ECC) approaches are usually considered in these scenarios.

While it is not shown in the figure, other devices could be communicated with gateway entities through proprietary protocols. Consequently, there is a real need for cryptographic schemes and security mechanisms that can be applied at upper communication layers, independently on the underlying protocols. In this sense, emerging approaches, such as CBOR Object Signing and Encryption (COSE) [15], represent significant efforts to apply advanced and optimized encryption and signature schemes to be considered.

3.1.3 Types of Devices and Messages

According to Figure 3.1, there are different components and entities in a typical IoT scenario. On the one hand, the gateways and the platform (and the services accessing to it) are usually considered as entities without resource constraints in which the application of conventional cryptographic schemes does not represent a major obstacle. As previously mentioned, in this case, the challenge is not related to the application of security protocols and cryptographic schemes, but with the need to define mechanisms manage the security and privacy aspects of large amounts of smart objects. In particular, there is a need to define approaches to bind security and privacy properties with semantic approaches, such as OMA Next Generation Services Interface (OMA NGSI) [16], so that applications and users do not need to know the details of the physical devices, but their virtual representation. This is particularly important for empowerment aspects, so users are enabled to define their security and access control preferences in a simple and scalable way.

On the other hand, smart objects represent the set of devices physically deployed in the environment that are responsible for capturing data from the environment and communicating it to other objects or services. They are usually represented as small devices with limited capacities that in many cases are made to fulfill a single simple task (e.g. measuring the temperature of a room). Furthermore, they are often featured by strong resource constraints in terms of processing power and memory. Specifically, Figure 3.3 shows the terminology considered for the definition of a constrained device to be used as a reference during the project.

In spite of these limitations, the application of cryptographic schemes in these devices is required in order to ensure that typical security properties, such as confidentiality or integrity, are maintained on an end-to-end

Name	data size (e.g., RAM)	code size (e.g., Flash)
Class 0, C0	<< 10 KiB	<< 100 KiB
Class 1, C1	~ 10 KiB	~ 100 KiB
Class 2, C2	~ 50 KiB	~ 250 KiB

Figure 3.3: Classes of Constrained Devices [1]

basis. In this sense, as in the previous subsection, the application of public-key cryptography is especially challenging because of the underlying mathematical operations. Additionally, it should be noted that a smart object may be required to produce data every certain period of time (e.g., 5 seconds), and therefore to protect such data accordingly. Consequently, there is a need to define a cryptographic approach so that devices do not exhaust their resources promptly.

3.2 Smart Building Use Case

Smart buildings represents an instantiation of the previous general setting, and a suitable scenario to demonstrate the applicability of the advanced cryptographic techniques and policy-based approaches to ensure users' empowerment for security and privacy. In these environments, large amounts of data need to be shared from heterogeneous devices to enable different services to make decisions accordingly; due to the amount and sensitivity of such information, users' privacy may be compromised if data protection schemes are not implemented. Specifically, the considered use case is based on a real scenario as part of an ongoing initiative at UMU premises, and derived from the SMARTIE project [17]. Figure **??** shows a simplified overview of the smart building use case by considering some of the already deployed components. Within this use case, two sub-cases are to be considered.

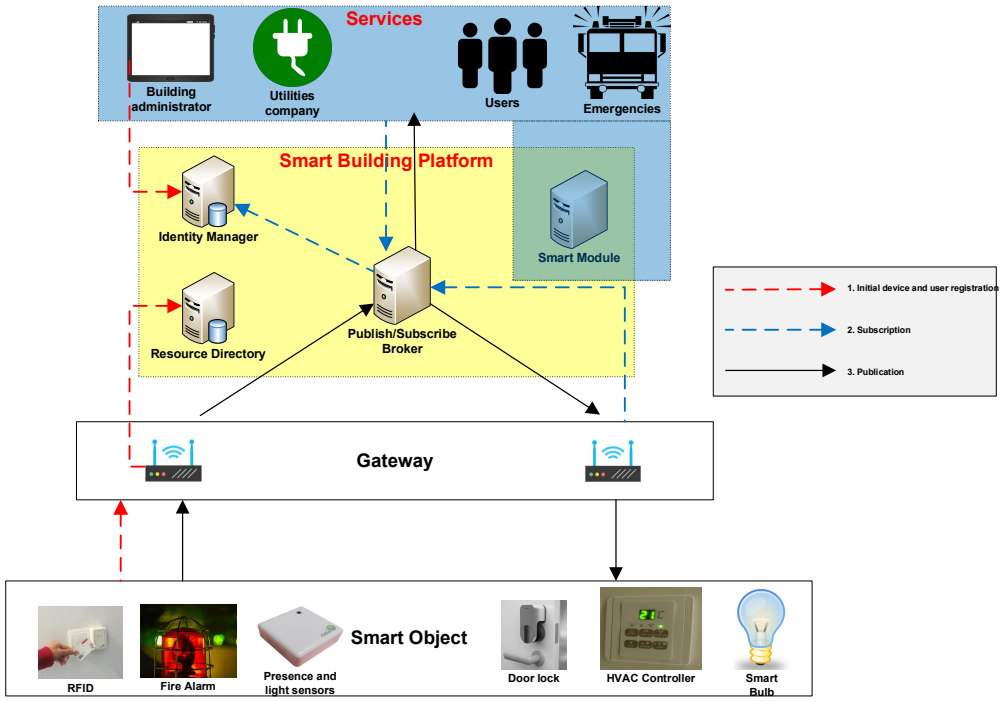

Figure 3.4: Smart Objects scenario overview

3.2.1 Case 1

The scenario is represented by a real smart building where a set of devices, such as smart meters, fire detectors, RFID readers, and other sensors are physically deployed. These appliances, including legacy devices, act as *Producer Smart Objects* and they are in charge of capturing the data associated with the daily activity in the building. Then, this information is sent to *Gateway* entities, which represent a central point for homogenizing the data communication from heterogeneous data sources to the *Smart Building Platform*. This platform represents a set of services and components that are intended to enable an efficient information exchange of large amounts of data from data sources to high-level *Services* and other devices (i.e. *Consumer Smart Objects*) through the *Publish/Subscribe Broker*.

In the building, an administrator is responsible for managing how devices' data in the building are accessed and shared. The management of these access control conditions may change frequently, for example due to the installation and uninstallation of new equipment for the building (R1). All building's smart objects must be registered in the *Resource Directory* entity that stores information of each device (e.g., its location), to ensure that only previously registered devices are able to publish data on the platform. In this sense, a previous *bootstrapping* process is assumed, so that each device has established a security association with the corresponding gateway. In addition, all building's users and services interested in building data are registered in the *Identity Manager* component, which stores their associated identity attributes (R2). Services and gateways are subscribed on the Publish/Subscribe Broker to be notified about information they are interested in. For the subscription process, this entity needs to be sure the requesting entity is already registered in the platform. The information that is used by services to offer additional functionality is provided by the smart objects through the gateways, therefore, the subscriber entities need data's authenticity and integrity (R3).

Each data type generated by smart objects can have different access conditions. For example, smart meters' energy consumption data could be only accessible by the building administrator and the utilities company, in order to monitor building energy use and generate a proper energy plan accordingly. However, these data should not be disclosed to other services or users, in order to avoid behavioral profiles can be inferred (R4). In addition to external services, the *Smart Module* represents the brain of the building and it is part of the Smart Building Platform. This service is responsible of optimizing its electric consumption by making decisions such as turning on/off the lights or the HVAC system depending on the number of users in a certain room. In addition, this service is in charge of detecting any suspicious behaviour of users in order to anticipate potential eventual incidents; for example, to detect if unauthorized users attempt to perform illegitimate activities on a building facility. In case of fire detectors' data, the building administrator and the emergencies services are responsible for managing a potential critical situation based on received data. This way, in a fire situation, some of users' data, such as their location and mobility condition need to be disclosed to additional services. This way, contextual conditions, such as a fire situation could suddenly change the access conditions to the information being sent to the broker (R5).

Beyond the use of the publish/subscribe broker, users and services can access other information and services already provided by the platform. Towards this end, they can use traditional login/password authentication, so they are unequivocally linked to their information within the Identity Manager. However, those users concerned about their privacy could also require the access to the platform in a privacy-preserving way, so only the personal information strictly necessary is to be disclosed (R6). Furthermore, users can become data producers, for example by using their smartphone or wearable devices.. For example, Alice could report an incident or damage to the building, so that the manager or administrator would be aware of it. However, Alice wants to prevent her behavior from being tracked so that the incident information does not disclose anything about her personal data (R7).

3.2.2 Case 2

The second case related to the smart building general use case is mainly taken from [18], Section 2.2 that has some aspects in common with the previous case. For this case, an IoT-enabled smart home is considered in which, like before, different sensors and actuator are deployed within it. Alice and Bob are the homeowners and they represent non-expert users who need to manage the access to their home devices (heating, lights, windows) by considering different types of users (such as friends, family, or the babysitter) (R9). In order to monitor the condition of their home, all devices can be remotely accessed by them. For example, they can increase the temperature from their work in a cold day, if they expect to be soon at home. However, only some of these intelligent objects can be accessed by other entities. For example, they want the data from smart meters to be accessed by the utility company, but not by neighbors or cleaning staff.

Alice and Bob have equipped their home with automated connected doorlocks and an alarm system at the door and the windows. The couple can control this system remotely. Alice and Bob have invited Alice's parents over for dinner, but are stuck in traffic and cannot arrive in time; whereas Alice's parents are using the subway and will arrive punctually. Alice calls her parents and offers to let them in remotely, so they can make themselves comfortable while waiting (R1). Then, Alice sets temporary permissions that allow them to open the door and shut down the alarm (R8). She wants these permissions to be only valid for the evening since she does not like it if her parents are able to enter the house as they see fit. When Alice's parents arrive at Alice and Bob's home, they use their smartphone to communicate with the door-lock and alarm system (R10). The permissions Alice issued to her parents only allow limited access to the house (e.g., opening the door, turning on the lights). Certain other functions, such as checking the footage from the surveillance cameras, are not accessible to them. Alice and Bob also issue similarly restricted permissions to e.g., cleaners, repairmen, or their nanny (R9).

3.2.3 Case 3

The third use case related to the smart building concerns advanced intrusion detection systems (IDSs) in that environment. The smart objects network deployed in the smart building is vulnerable to multiple attacks aimed to disrupt the network. The use of cryptographic primitives, e.g., attribute-based encryption, to provide preventive security in the network fails to secure the network against some types of attacks whereby a smart object is cloned or compromised. A malicious smart object owning legitimate cryptographic keys can easily launch an internal attack inside the network. At the network level, it can maliciously drop packets, delay the transmission, or send packets through a different route than planned, whereas at the application level, it can send bogus information to authorized recipients or violate the access policy associated with encrypted data by adding attributes of malicious ones. Such an internal attacker can only be detected through behavioral analysis mechanisms that track unusual smart objects behavior, the network activity and interactions between objects to detect threat attempts and/or occurrences. Once a security anomaly is detected, a reaction mechanism is launched to take repair measures.

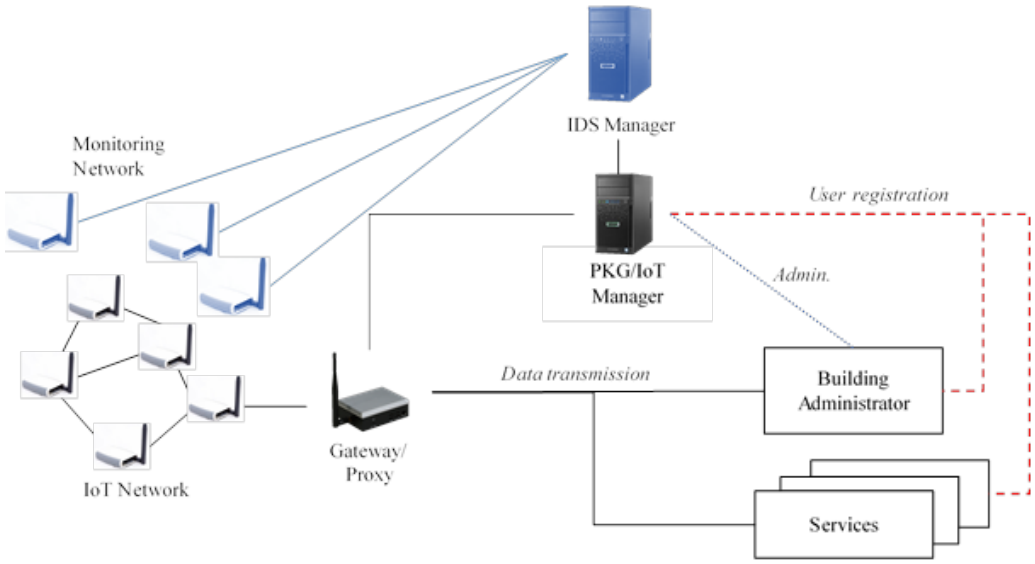

Figure 3.5: Diagram for Case 3

3.3 Requirements

3.3.1 General Security Requirements and Threats

The Smart Building scenario highlights a set of security and privacy challenges that are more specific to IoT environments, beyond typical Web scenarios. Firstly, to realize and implement a holistic security and privacy approach for a data-centric IoT, scalable and flexible cryptographic mechanisms must be applied, beyond the use of well-known and established transport security protocols (e.g. (D)TLS)). In this sense, as already mentioned there is a need towards the application of data-centric (or object-centric) security mechanisms for IoT, as demonstrated recently by the IETF with the COSE WGs. Indeed, the protection of data itself is crucial to ensure that only authorized entities access such data, and that these are produced by legitimate and trustworthy entities. Secondly, related to the previous aspect, the proposed scenario (especially the case 1) highlights the need for a non-interactive security and privacy model in which communicating parties may not be directly interacting directly. This is especially relevant in typical IoT scenarios where data need to be shared with groups of users or services, or when interacting entities are decoupled (e.g. through the use of a central platform). In this sense, the application of flexible and scalable encryption and signature algorithms is key in order to guarantee the adequacy of a non-interactive model. However, its realization must cope with the requirements imposed by the underlying cryptographic schemes. On the one hand, while symmetric key approaches can be tailored to small or specific scenarios, its scalability (e.g. regarding key management) and flexibility makes its adoption inappropriate. On the other hand, the application of public key schemes could be infeasible especially in the case of resource-constrained devices and networks. In this sense, there is a real need to consider performance (e.g. execution time or power consumption) and bandwidth (e.g. due to key and ciphertext size) issues, which are imposed by many common IoT scenarios.

By considering the previous scenarios, the main threats to be considered are:

- Lack of Authentication. All the entities interacting with the platform must be properly authenticated, and allowing privacy preservation when required.
- Lack of Authorization. Smart objects and their data must be only accessible to entities with the rights to do it.
- Denial of Service attacks. Given their constraints, smart objects could be especially vulnerable to these attacks, so the cryptographic approach must take into account this aspect.

- Replay attacks. Even when data go through intermediate entities, replay attacks must be detected.
- Insufficient Cryptography. The cryptographic suite and key length must be enough according to NIST recommendations
- Lack of confidentiality. Data need to be confidentiality protected on an end-to-end basis, so it cannot be accessed by unintended entities even when going through intermediate entities, such as gateways
- Lack of integrity. Data need to be integrity protected on an end-to-end basis, so it cannot be modified in transit even when going through intermediate entities, such as gateways

3.3.2 Specific Requirements

In addition to the generic requirements and threats from the previous subsection, the specific requirements for the smart building use case can summarized as:

- R1. The owner of a smart object in the building requires that the policies defined for accessing her data can be dynamically changed.
- R2. Any entity that interacts with the platform must be previously registered.
- R3. Any service or entity subscribed to the broker require the integrity and authenticity of the data coming from smart objects, even if such data go through gateways.
- R4. The owner of a smart object in the building requires the confidentiality of their data, so that they are not disclosed to unauthorized services or entities.
- R5. The owner of a smart object in the building requires that the policies defined for access to their data take into account the current context.
- R6. Any service or users can access the platform components in a privacy-preserving way.
- R7. A user could report an incident or damage in the building, but she requires to maintain her privacy, while integrity of the information is still ensured.
- R8. The owner of the building devices may require temporary permission to other users or services to access their devices.
- R9. The owner of a smart object in the building can grant different permissions to different types of users (for example, family, friends or nanny)
- R10. The smart object needs to enforce the access control policies defined by the owner.

The mentioned requirements and aspects related to this use case are summarized in the following table.

ID	Type	Priority	Description	Rationale	Dependencies
SOREQ1	NFREQ	MUST	Producer's data integrity must be supported	To ensure producer's data is not modified	-
SOREQ2	NFREQ	MUST	Producer's data confidentiality must be supported	To ensure producer's data is only accessible for intended entities	-
SOREQ3	NFREQ	MUST	Obtaining cryptographic keys and credentials requires strong authentication	Only legitimate entities are able to obtain corresponding credentials	-
SOREQ4	NFREQ	MUST	Secure key storage	Cryptographic keys should be securely stored to avoid attackers to discover or modify their value	-
SOREQ5	NFREQ	SHOULD	Secure signature (verification) outsourcing	In case of resource-demanding signature algorithms, an assistant gateway will not be able to obtain smart objects' private key or message	SOREQ1
SOREQ6	NFREQ	SHOULD	Secure encryption/decryption outsourcing	In case of resource-demanding encryption/decryption algorithms, a assistant gateway will not be able to obtain smart objects' private key or message	SOREQ2
SOREQ7	NFREQ	MAY	Distributed cryptographic outsourcing	In case of resource-demanding cryptographic algorithms, a group of assistant gateways could collaborate to perform the required operations	SOREQ5, SOREQ6
SOREQ8	NFREQ	MUST	Collusion resistance	Different smart objects will not be able to access a certain information, even though they combine their cryptographic keys	SOREQ3, SOREQ4
SOREQ9	NFREQ	SHOULD	Use of scalable cryptographic algorithms beyond the use of symmetric-key cryptography approaches	Ensure flexibility and scalability of the cryptographic approach	SOREQ1, SOREQ2
SOREQ10	NFREQ	SHOULD	Use of privacy-preserving signature algorithms	A producer smart object could ensure data integrity while its real identity is not disclosed	SOREQ1
SOREQ11	NFREQ	SHOULD	Use of flexible encryption algorithms	A producer smart object could ensure data confidentiality with groups of smart objects and/or services	SOREQ2
SOREQ12	NFREQ	MUST	Policy-based approaches for defining security and privacy preferences	Users need to be empowered for defining how their smart objects' data are shared with other entities	-
SOREQ14	NFREQ	MAY	Use of transport layer security mechanisms	To complement cryptographic algorithms in cases in which entities are known or they must interact	SOREQ1, SOREQ2
SOREQ15	NFREQ	MUST	Access to platform's data or services will be protected	Platform's data and services must be only accesible to legitimate and authorized entities	-
SOREQ16	FREQ	MUST	Access to platform's data or services could be done in a privacy-preserving way	Users could desire preserve their privacy when accessing platform's services	-
SOREQ17	FREQ	MUST	The platform must allow publish/subscribe interactions	Interacting smart objects or entities could be decoupled	-
SOREQ18	FREQ	SHOULD	Information and semantics models should be supported by the platform	Access to the platform should be homogenized to deal with heterogeneous smart objects	-
SOREQ19	FREQ	MAY	The platform must provide storage facilities	Large amounts of data could be required tfor further processing	-
SOREQ20	FREQ	MAY	The platform must provide analytic facilities	Information can be infered from data received from different smart objects	-

4 Initial Considerations

The set of requirements provided in the previous sections can be divided in two main complementary paths: empowerment mechanisms for security and privacy, and the integration of cryptographic algorithms and resulting protocols on constrained smart objects and networks.

Initially, for the definition of users' empowerment mechanisms, eXtensible Access Control Markup Language (XACML) [19] is considered as a consolidated approach to help for defining access control preferences. XACML is widely used nowadays on typical web scenarios, but it has been rarely considered for the IoT setting. For the smart objects setting, XACML can be instantiated through the definition of components in the platform, for the definition and decision of access control policies. For example, the Policy Administration Point (PAP) and the Policy Decision Point (PDP) can be defined as new entities, and integrated with the Identity Manager and the Resource Directory, so access control decisions can be made based on identity attributes. In addition, it can be combined with a token-based authorization approach so that users can set temporary permissions on their objects, and these devices are able to enforce access control decisions. As already described in Section 3.2.2, this is especially relevant in everyday situations in which access rights need to be temporary granted. In addition, emerging approaches such as MyData [20] may also be considered to foster compliance with user consent aspects , in order to help for governing the disclosure of personal data.

In the case of the integration of cryptographic mechanisms into constrained devices, USEIT aims to integrate cryptographic algorithms to address interactive as well as the already mentioned *non-interactive model*. Beyond technical issues, such as performance and bandwidth aspects, the cryptographic schemes to be considered in this use case must take into account the mentioned aspects regarding scalability and flexibility. On the one hand, given the potentially high number of smart objects in many use cases, key management and revocation tasks should be reduced as much as possible. On the other hand, the proposed schemes should be flexible enough to be accommodated for in different scenarios, devices and considering advanced identity management schemes (e.g., based on identity attributes).

In this sense, encryption algorithms represent an essential component to ensure only legitimate and authorized entities will be able to access the data. Beyond the use of traditional approaches based on symmetric and public-key cryptography, Attribute-Based Encryption (ABE) schemes are receiving increasing attention due to their high level of flexibility and expressiveness. In these schemes, entities are represented by identity attributes, so that data will be accessible only to participants satisfying specific combinations or sets of attributes. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [21] is is a variant of ABE whereby data are encrypted under a logical combination of identity attributes (access policy), while private keys are associated with a set of attributes. Therefore, data will be decrypted only by those entities whose private keys satisfy the conditions specified in the access policy. However, it requires extensive processing capabilities to execute highly resource-demanding cryptographic operations. This is especially relevant in scenarios where huge amounts of data will be exchanged and involving resource constrained devices, since scalability and applicability can be limited. Furthermore, end-to-end confidentiality may be compromised if encryption/decryption algorithms cannot be accommodated on end-devices. As an example of it, Figure 4.1 shows the delay required to decrypt a ciphertext according to the number of attributes that were used to encrypt. These results were obtained with a MIPS 32bits, 64MHz, 512Kb Flash and 128Kb RAM device, and the cpabe toolkit of the PBC library [**?**].

To complement the use of encryption algorithms, the application of signature schemes to the data produced by smart objects is crucial to ensure integrity. In this sense, in addition to well-known schemes based on public-key cryptography, the use of anonymous signatures [22] can be considered as a privacy-preserving approach to realize a security and privacy attribute-based solution.

Unlike typical approaches in which constrained smart objects delegate heavy cryptographic operations into

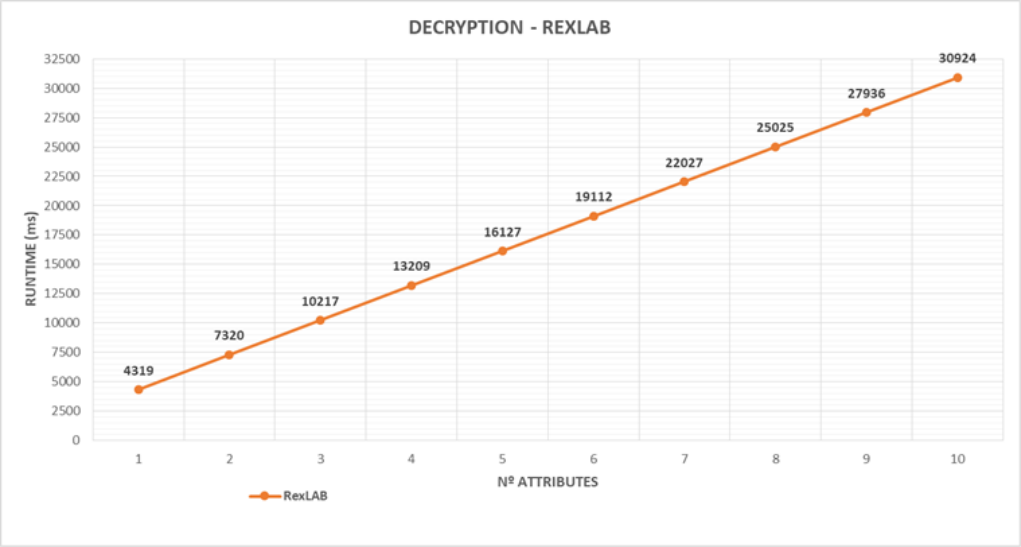

Figure 4.1: Example of CP-ABE results (Decryption)

powerful devices, USEIT will analyze an alternative approach in order to avoid these entities become into security and privacy bottleneck. In a naive setting, smart objects' private keys are given to more powerful entities so they can sign or decrypt a ciphertext on behalf of constrained devices. In this sense, approaches based on threshold cryptography [23] could be considered as the baseline, so cryptographic operations can be split into a group of such devices. In threshold cryptography, at least a threshold number of entities are required to perform a cryptographic operation, and consequently, it could help to mitigate the issues associated to common centralized approaches.

5 Conclusions

This document summarizes the requirements of a number of areas that will for the main points of attention for the USEIT project.

One main focus of USEIT will be V2V and V2I communication in a C-ITS scenario. The main challenges here are posed by the stringent limitations in terms of bandwidth and connectivity, combined with the highly privacy-sensitive nature of location data.

Another main focus will be on smart objects in a Smart Building Platform with connect devices such as smart meters, fire detectors, connected doorlocks, and other sensors. Application scenarios range from industrial buildings to private homes. USEIT will address the challenges of meeting the highly flexible and dynamic nature of these networks, combined with the low computational power of many of the connected devices.

Bibliography

[1] C. Bormann, M. Ersue, and A. Keranen, "RFC 7228 - Terminology for Constrained-Node Networks)," 2014.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[4] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the internet of things," *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, 2010.

[5] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[6] A. Ruddle, D. Ward, B. Weyl, S. Idrees, Y. Roudier, M. Friedewald, T. Leimbach, A. Fuchs, S. Gürgens, O. Henniger, R. Rieke, M. Ritscher, H. Broberg, L. Apvrille, R. Pacalet, and G. Pedroza, "Security requirements for automotive on-board networks based on dark-side scenarios," EVITA Project, Deliverable D2.3, 2009.

[7] R. Kroh, A. Kung, and F. Kargl, "Vanets security requirements," SEVECOM Project, Deliverable D1.1, 2006.

[8] M. Mattheß, N. Bißmeyer, J. Schütte, J. P. Stotz, M. Gerlach, F. Friederici, C. Sommer, H. Seudié, W. Stephan, E. Hildebrandt, J. Vogt, B. Allani, T. Gansen, A. Jentzsch, H. Stübing, and A. Jaeger, "Spezifikation der it-sicherheitslösung," simTD Project, D21.5, 2009.

[9] T. Benz, A. Kung, M. Kost, F. Kargl, Z. Ma, G. Tijskens, and J. Freytag, "V2z privacy issue analysis," PRECIOSA Project, Deliverable D1, 2009.

[10] J. P. Stotz, N. Bißmeyer, F. Kargl, S. Dietzel, P. Papadimitratos, and C. Schleiffer, "Security requirements of vehicle security architecture," PRESERVE Project, Deliverable D1.1, 2011.

[11] ETSI, "Intelligent transport systems (its); security; threat, vulnerability and risk analysis (tvra)," TR 102 893, 2010.

[12] O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle, and B. Weyl, "Security requirements for automotive on-board networks," in *9th International Conference on Intelligent Transport System Telecommunications (ITST 2009)*, 2009.

[13] Wikipedia, "Braking distance," 2018. [Online]. Available: https://en.wikipedia.org/wiki/Braking_distance

[14] Z. Shelby, K. Hartke, and C. Bormann, "RFC 7252 - The Constrained Application Protocol (CoAP)," 2014. [Online]. Available: http://dx.doi.org/10.17487/rfc7252

[15] J. Schaad, "RFC 8152 - CBOR Object Signing and Encryption (COSE)," 2017.

[16] M. Bauer, E. Kovacs, A. Schulke, N. Ito, C. Criminisi, L. Goix, and M. Valla, "The Context API in the OMA Next Generation Service Interface," in *2010 14th International Conference on Intelligence in Next Generation Networks*. Institute of Electrical & Electronics Engineers (IEEE), 2010. [Online]. Available: http://dx.doi.org/10.1109/icin.2010.5640931

[17] SMARTIE, "Deliverable 2.2: Smartie requirements," 2015. [Online]. Available: http://www.smartie-project.eu/download/D2.2-Requirements.pdf

[18] L. Seitz, S. Gerdes, G. Selander, M. Mani, and S. Kumar, "RFC 7744 - Use Cases for Authentication and Authorization in Constrained Environments," 2016.

[19] T. Moses *et al.*, "Extensible access control markup language (xacml) version 2.0," *Oasis Standard*, vol. 200502, 2005.

[20] A. Poikola, K. Kuikkaniemi, and H. Honko, "Mydata a nordic model for human-centered personal data management and processing," *Finnish Ministry of Transport and Communications*, 2015.

[21] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007, pp. 321–334.

[22] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 21–30.

[23] Y. Desmedt, "Threshold cryptosystems," in *International Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1992, pp. 1–14.