CHIST-ERA

User empowerment for SEcurity and privacy in Internet of Things

Final requirements of Use Cases

Deliverable number: D1.4

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:	Antonio Skarmeta, Universidad de Murcia (UMU) – Spain
Deliverable nature:	Report
Dissemination level:	Public
Delivery Date:	November 2018
Authors:	Salvador Pérez, University of Murcia Jorge Gallego, University of Murcia Ramón Sánchez University of Murcia Pedro González-Gil, University of Murcia Antonio Skarmeta, University of Murcia Jan Camenisch, work done while at IBM Research – Zurich Manu Drijvers, work done while at IBM Research – Zurich Anja Lehmann, IBM Research – Zurich Gregory Neven, work done while at IBM Research – Zurich Patrick Towa, IBM Research – Zurich

Abstract

This document provides security solutions to be deployed on different IoT-enabled scenarios, with the aim of fulfilling the requirements identified in such scenarios. Specifically, two security solutions are proposed: the first one takes as starting point the C-ITS framework and introduces zone encryption as a means to improve security and privacy in V2V communications; the second one is based on the Smart Objects framework and presents a security CP-ABE-based architecture that makes use of certain FI-WARE enablers with the aim of providing an interoperable mechanism to share data between different groups of entities in a secure way.

Contents

Executive Summary

USEIT wants to use the experience on previous results from projects, such as ABC4Trust, FutureID, Sociotal, Smartie, ITTSv6, and others, and focus on the identification of the security and privacy components needed to extend and support the objectives of USEIT vision.

1 Introduction

The present deliverable is intended to provide certain security proposals based on the proposed C-ITS and Smart objects frameworks, both described in D3.2. It should be pointed out that such proposals also consider security requirements introduced in D1.3. According to it, we will focus on two different proposals. In a first part, we first recall the main challenges of vehicle-to-vehicle (V2V) communication and describe the solutions that the EU and US C-ITSs currently intend to adopt. We explain how these solutions only partially address the V2V challenges, and then we introduce zone encryption schemes to improve the security and privacy of V2V communications. In a second part, we present a security proposal makes use of attribute-based encryption techniques and certain FI-WARE enablers to enable both data protection according to users' preferences, and group data sharing. Note that the proposal is intended to be deployed in real IoT scenarios.

1.1 Related deliverables

This deliverable considers other work provided in the following deliverables:

- D1.3 describes the major security/privacy requirements extracted from different IoT use cases where the C-ITS and Smart object frameworks may be deployed.
- D3.2 defines the C-ITS and Smart objects frameworks, identifying the main entities and phases defined to carry out their functionality.

1.2 Deliverable outline

The document consists of two technical chapters:

- Chapter 2 first describes the requirements for V2V communication systems, as well as the solutions (which do not fulfill all these requirements) currently proposed for the EU and US C-ITSs. It then introduces zone encryption as a suitable and efficient solution for V2V communication with strong security and privacy guarantees.
- Chapter 3 presents the Smart objects framework-based proposal that aims to enable security during group data sharing by considering users' preferences.

2 Privacy-Preserving V2X Communication

Vehicle-to-Vehicle (V2V) communication systems are currently being prepared for real-world deployment as early as 2019, but they face strong opposition over privacy issues. Vehicle position beacon messages are the main culprit, as they are planned to be broadcast in clear text. So far no practical solutions have been proposed to encrypt or anonymously authenticate V2V messages.

2.1 Challenges in V2X Communication

V2V communication and Vehicle-to-Infrastructure (V2I) communication, together known as V2X, mainly consist of two types of messages: occasional event-triggered safety messages (e.g., emergency braking maneuver) and regular position beacon messages. The latter category, known in the European Cooperative Intelligent Transport Systems (C-ITSs) as Cooperative Awareness Messages (CAMs), carry dynamic information about the sending vehicle such as its position, speed and heading, as well as (semi-)static information such as the vehicle dimensions and its sensor accuracy. CAMs are primarily broadcast over an unprotected short-range radio channel (ETSI ITS-G5), and are thus easy to intercept and they potentially leak sensitive information about people's whereabouts, travel itineraries, and driving habits. Concrete threats include burglars tracking houses left unoccupied, stalker following their victims from an out-of-sight location, and mass surveillance of entire cities at an amortized cost of dollar-cents per vehicle per year. Besides, privacy regulations forbidding misuse of CAM data are difficult to enforce as rogue eavesdropping devices are easy to build and nearly impossible to detect, let alone localize.

2.1.1 Privacy and Authentication in V2X

Due to the open nature of C-ITSs and the problems of managing encryption keys among constantly changing groups of vehicles, encryption in V2X has mostly been considered prohibitive. Encrypting CAM data seems an obvious choice, but doing so in a practical and useful manner is not straightforward. The necessarily open nature of C-ITSs requires that all nearby vehicles can decrypt. Embedding the same symmetric key in all units is not feasible as no revocation is possible when the key gets compromised. In this respect, public-key encryption is better since the receiving device has to make itself known to the senders so that the senders know which public key to encrypt to. However, bandwidth restrictions prohibit one-to-one connections, and the high frequency of CAMs does not allow them to be large enough to include a separate ciphertext for each receiving vehicle.

Possible better solutions such as multi-sender broadcast encryption [1] or public-key traitor tracing [2, 3] do not scale to a setting with hundreds of millions of vehicles. Furthermore, as for symmetric encryption, used alone, these solutions make it impossible to localize or even detect rogue devices eavesdropping on the V2X communication.

Most research in security and privacy for V2X has consequently focused on the authentication aspect ensuring that messages originate from genuine vehicles without making individual vehicles traceable throughout the system. The work of Petit et al. [4] offers an excellent survey of this field.

2.1.2 Efficiency Requirements

In a C-ITS, each vehicle is expected to broadcast beacon messages 1–10 times per second, and process up to 3000 incoming CAMs per second. On this ground, the EU and US C-ITSs have imposed stringent restrictions on the size of the messages exchange between vehicles: each CAM should not carry more than 300 Bytes of cryptographic overhead, the cryptographic operations must be fast and preferably, the storage cost must be as low as possible.

2.2 Current Solution

The practical C-ITS systems that are currently considered for deployment in Europe and the US both take a similar approach to authentication by letting vehicles sign outgoing messages with short-lived pseudonym certificates. They provide some degree of privacy by making vehicles frequently change their certificates from a small pool of pseudonyms. In the European C-ITS, the vehicles have to periodically download and store irrevocable certificates (around 100 pseudonyms per vehicle per week) from a certification authority. In the US C-ITS, vehicles come preloaded with three years' worth of revocable pseudonym certificates.

These solutions with pseudonyms however enforce a trade-off between security, privacy and efficiency. A larger pseudonym pool size guarantees more privacy, but is expensive to download and store. It also provides less protection against Sybil attacks in which an attacker impersonate several vehicles at once by compromising a single vehicle. The compromises of a hundred pseudonyms per vehicle per week in the EU thus combines mediocre privacy guarantees for frequent drivers, high storage costs and practically no resistance against Sybil attacks. The US solution suffers from even bigger storage costs, worse resistance against Sybil attacks and, depending on the frequency of use, weak privacy guarantees.

Moreover, with just pseudonyms, even if the CAMs are authenticated, they are still sent in clear text. It therefore makes it possible for anyone, from anywhere, to track any vehicle just by recording the conversation among the vehicles. The current solutions thereby provide a mass surveillance tool anybody, and on top of that at a negligible cost.

Designing an efficient solution that offers encryption of CAMs, strong security and privacy guarantees and the possible revocation of rogue devices is then still an open problem.

2.3 Zone Encryption with Anonymous Authentication

We propose a novel solution for V2X communication that does not only provide strong security and privacy guarantees, but also allows for the encryption of CAMs and enables the revocation of rogues vehicles. It consists of a novel mechanism called *zone encryption* and that allows vehicles to authentically and confidentially send CAMs to each other *without the use of any infrastructure*. Zone encryption is therefore a purely V2V (rather than V2X) communication system.

2.3.1 Description of Zone Encryption

With zone encryption, a vehicle securely communicates with the other vehicles in its *immediate vicinity* at a given time, encrypting all CAMs. The main insight of zone encryption is precisely to leverage the fact that only vehicles that are close to each other need to communicate. The road network is then assumed to be sub-divided in *zones*, and the vehicles within a given zone can agree on a shared symmetric encryption key. For example, the zone boundaries could be derived statically from the GPS coordinates and be chosen so that the longest straight line distance within a zone is less than the transmission radius of the radio signal (typically 300–500m). Doing so ensures that any two vehicles in a zone can communicate. However, it should avoided that two vehicles that are physically close but at opposite sides of a zone boundary cannot communicate because they broadcast CAMs encrypted for different zones. On this ground, zone encryption lets vehicles simultaneously broadcast CAMs encrypted for multiple zones.

To obtain a zone key, a vehicle must *anonymously authenticate* itself to the other vehicles in the zone. To do so, each vehicle is assigned a long term credential by an *enrollment authority*, and using it, must periodically (e.g., every week) obtain short-term credentials from an *issuance authority* (or simply *issuer*). Nevertheless, in case a suspicious behavior is detected, the issuer can potentially revoke the anonymity of a vehicle and its long term credential.

We however impose that zone keys are frequently refreshed, e.g., every 15 minutes. This ensures that a rogue eavesdropping device cannot simply stay silent and listen to ongoing traffic, but has to interact actively with other vehicles by sending key requests or responses, exposing itself to detection and localization through triangulation. Via an opening algorithm run by the issuer, the long-term credential of a detected rogue decryption device can be traced and revoked. This does not only disable the decryption capabilities of the detected device, but also of any other rogue devices based on the same compromised key. Mass production of such rogue devices thereby becomes much more expensive, potentially beyond the point of economic feasibility.

2.3.2 Security and Privacy of Zone Encryption

To use a zone encryption scheme as a deployed V2V communication system, it should guarantee the following security and privacy properties:

- Confidentiality of encrypted CAMs: a zone encryption scheme must guarantee that no attacker can infer any information about the CAMs encrypted for a zone at a given time unless it has obtained the zone key at that time.
- Anonymity of vehicles: anonymity, main privacy requirement of V2V communication, ensures that the encrypted CAMs and the messages sent during key requests and responses carry no information about the sending vehicle.
- Only legitimate Vehicles can send valid CAM ciphertexts: this property consists of two sub-properties:
  - Traceability of vehicles: which means that if a vehicle knows a zone key at a given time, then it must have requested it by sending a message that can be traced by the issuer to its long-term credential. The issuer can then revoke the long-term credential if necessary.
  - Integrity of encrypted CAMs: which guarantees that no attacker can compute a ciphertext that is considered as a valid CAM encryption by the vehicles in a zone at a given time if the attacker does not know the zone key at that time.

  Combined, these two sub-properties ensure that only the vehicles that have a short-term credential and a zone key can send encrypted CAMs that are accepted by the other vehicles in that zone. They thereby guarantee the authenticity of the CAMs exchanged between vehicles.
- Sybil Resistance: an attacker should not be able to impersonate several vehicles at once by compromising a single vehicle. Note that this property lies on the authentication mechanism used by vehicles during key requests and responses achieves as it is only during key requests and responses that credentials (each of which is tied is tied to a specific vehicle) are used.

A zone encryption scheme that satisfies these properties can then be used a V2V communication system which offers encryption of CAMs, strong security and privacy guarantees and the possible revocation of compromised devices.

2.3.3 Efficiency of Zone Encryption

Zone encryption combines shared zone keys and anonymous authentication between vehicles with short-term credentials. As a result, it circumvents both the inefficiency of having an asymmetric pair of keys for each vehicle (and the issue of not being able to revoke rogue vehicles with just symmetric encryption). From an efficiency perspective, it means that by encrypting a CAM for a zone, a vehicle privately send it to *all* the vehicles in that zone. Since a vehicle must only communicate with the vehicles in its zone and the ones surrounding it, each CAM should carry little cryptographic overhead, well within the 300 Byte limitation.

Furthermore, zone keys and anonymous authentication should translate in low storage costs as each vehicle need only store at a time a long-term, a short-term credential and the keys for its zone and the surrounding ones. It thus achieves a important improvement compared to the US proposal which advocates downloading and storing three years' worth of pseudonym certificates.

Notice also that during its period of validity (typically one week), a short term credential also equates to an unlimited amount of pseudonyms as a vehicle can unrestrictedly use it to authenticate itself to other vehicles. It constitutes a significant improvement in terms of communication with issuers compared to the current EU proposal in which vehicles have to spread out requests for individual pseudonyms over time, rather than downloading them in batches, to avoid that issuers are able to link the pseudonyms belonging to the same vehicle.

As for speed performances, since symmetric cryptography is typically faster than its asymmetric counterpart, the use of symmetric zone keys should also incur an efficiency improvement compared to the current EU and US proposals that propose to pseudonymously sign and verify each CAM.

Zone encryption therefore offers an efficient V2V communication system, an even more efficient one that the current EU and US proposals in some regards, although it guarantees stronger privacy and security (including encryption of CAMs) than they do.

3 Protecting personal data in IoT platform scenarios through encryption-based selective disclosure

As the Internet of Things evolves, citizens are starting to change the way they share information and communicate with their surrounding environment, enabling a constant, invisible and sometimes unintended information exchange. This trend raises new challenges regarding user's privacy and personal consent about the disclosure of personal data that must be addressed by flexible and scalable mechanisms. Towards this end, this work introduces the concept of bubble, as a coalition or group of smart objects that can be created according to the relationship between their owners. The proposed approach is based on the use of attribute-based encryption to protect the associated data according to users' preferences, and FI-WARE components for deployment purposes. As a scenario example, the solution is integrated with a radio localization system, in order to protect location data in the context of smart buildings. Finally, this work provides implementation details about the required components, as well as their evaluation on real smart environment scenarios.

3.1 Introduction

The inclusion of *Information and Communications Technology* (ICT) in our everyday environments has motivated the development of different initiatives to encourage the deployment of new services and applications in the scope of Smart Cities [5]. Much of the success of these scenarios is based on the continuous data sharing from a huge amount of heterogeneous data sources, such as smartphones, transport infrastructures or devices physically deployed in our surrounding environment. These devices are currently enabled to share their sensed data to platforms that extract knowledge from that information, in order to provide more sustainable and efficient cities to citizens.

In this sense, trust perception of users will be key for the success of new services and applications. The volume and sensitivity of the information to be exchanged will require effective and automated mechanisms to make users aware about how their data are shared. Given the huge number of devices producing data, as well as the number of services and citizens that will make use of this information, the application of scalable mechanisms to protect the access to shared information is crucial. Furthermore, such mechanisms should make use of cryptographic approaches that do not require complex key management tasks. Indeed, protecting the access to highly sensitive information is a challenging aspect in scenarios where a large number of heterogeneous devices need to interact, and where different actors are involved with conflicting interests. On the one hand, companies require a huge amount of information, in order to provide rich experiences and customized services to users. On the other hand, for citizens, maintaining the control on how their data are shared and under what circumstances is a difficult goal to be accomplished.

In order to address these challenges, this work presents the concept of *bubble*, as well as its design and implementation by using specific technologies and mechanisms. A bubble can be defined as a group of smart objects whose associated data are shared under a set of security restrictions according to users' preferences. The proposed approach is based on the use of *Attribute-Based Encryption* (ABE) [6], which provides a high level of flexibility and scalability to address some of the requirements for providing confidentiality to shared data. Specifically, we have made use of the *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE) scheme [7], in which each piece of information is encrypted under a certain logic combination (or policy) of attributes, while a private key is associated with a certain set of these attributes. Thus, an entity is able to decrypt a certain ciphertext if its key satisfies the policy that was used to encrypt it. CP-ABE provides two main properties to be leveraged in these scenarios. On the one hand, unlike symmetric key cryptography, it does not require entities to share cryptographic keys, which could be unfeasible given the huge number of data producers and consumers.

On the other hand, it is seamlessly applicable to maintain confidentiality in one-to-many configurations (even in the presence of a central platform), since a piece of information can be encrypted and shared with groups of entities using a single message. The resulting approach has been deployed on the FI-WARE platform[1] through the integration with the *Orion Context Broker Generic Enabler* (GE)[2], which is intended to serve as a central platform for sharing information among different entities, and an extension of the IdM Keyrock proposed in [8]. The integration of the CP-ABE scheme into this platform allows to leverage the features already offered by existing GEs, such as the use of the publish/subscribe pattern, and the *OMA Next Generation Services Interface* (OMA-NGSI) [9] as the data format to represent information in such sharing ecosystem. Furthermore, the performance of the resulting approach has been evaluated on a real scenario by considering different software components and practical evaluation aspects. Based on these aspects, the main contributions of this work are:

- Definition of *bubble* as a concept to represent groups of entities, whose data are protected under different combination of attributes by considering potential relationships between their owners.
- Use of CP-ABE to enforce data protection, and integration with existing IdM schemes and tools to foster the realization of an attribute-based data sharing ecosystem.
- Implementation and integration with current and enriched FI-WARE components to promote compatibility and interoperability with existing IoT deployments.
- Implementation of open-source prototypes[3] as part of the EU FP7 project SOCIOTAL[4], as well as an exhaustive performance evaluation by using different settings and components.

The remainder of this paper is organized as follows: Section 3.2 presents a description of some works related to our proposal. Section 3.3 describes the main building blocks of our approach. An overview of the required architecture and main interactions among the identified components is given in Section 3.4. Such overview is detailed in Section 3.5 where some usage examples are also provided. The integration of the resulting approach with a radio localization system is described in Section 3.6 to show the applicability of the proposal on a real scenario. Section 3.7 describes the implementation details of the required components, and a exhaustive performance evaluation is given in Section 3.8. Finally, Section 3.9 concludes the paper with an outlook of our future work in this area.

3.2 Related Work

In recent years, the security challenges and requirements that are derived from common scenarios and use cases in smart cities have attracted significant interest from academia [10]. The huge volume and continuous data exchange, mainly fostered by the strong development of IoT, as well as the sensitivity of the information being exchanged, make users' privacy more difficult to be preserved. Indeed, data protection represents one of the main cornerstones to cope with security and privacy concerns in IoT scenarios [11] [12]. This is also significantly reflected by the application of the *General Data Protection Regulation* (GDPR)[5] as the main legal framework governing data protection aspects in Europe.

In common IoT-enabled scenarios, physically deployed devices (e.g. a sensor) senses and processes data about its surrounding environment, which are shared with groups of users or applications through central data platform. This component is additionally intended to apply data mining techniques [13] to extract knowledge from the information that is transmitted from data sources. However, since it will have access to all the information, this platform becomes an obstacle to ensure that only authorized entities are able to access the information being shared, which requires to move the enforcement of security preferences closer to data sources. In this

[1] http://www.fiware.org/

[2] http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker

[3] https://github.com/sociotal?tab=repositories

[4] http://sociotal.eu/

[5] https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en

sense, the application of flexible and scalable encryption schemes is crucial to ensure only authorized users and applications will have access to shared data.

In parallel with the strong development of data-driven smart cities, *Attribute-Based Encryption* (ABE) [6] has received notable attention, due to its flexibility to provide confidentiality on data outsourcing scenarios, and their advantages compared to the use of symmetric key cryptography [14]. ABE represents the generalization of *Identity-Based Encryption* (IBE), in which users' identities are represented as a set of attributes (e.g. nationality, gender or address). Based on ABE, two alternative schemes were proposed. On the one hand, in the *Key-Policy Attribute-Based Encryption* (KP-ABE) [15], a piece of data is encrypted with a set or list of attributes, while private keys are associated with logical combinations of attributes. On the other hand, in a *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE) scheme [7], data are encrypted under a policy of attributes, while private keys of participants are associated with sets of attributes. This way, the entity responsible for outsourcing a certain piece of information can exercise control over how their data are shared by specifying the combination of attributes that must be satisfied by intended receivers. Currently, there are two main lines of research related to the use of CP-ABE. Firstly, the direct application of CP-ABE presents some practical challenges related to the size of the ciphertexts and keys and the use of expensive cryptographic primitives that are required by the scheme. These limitations have led to the emergence of different theoretical works [16] [17], in order to provide more lightweight, expressive and practical alternatives. Secondly, the advantages of CP-ABE are being exploited in different use cases related to the Cloud Computing paradigm [18] [19] [20]. These approaches partially fulfill some of the security needs to be addressed in IoT scenarios where information needs to be shared through central data platforms. Furthermore, they do not consider the inclusion of IoT devices acting as data sources, which are usually deployed on smart cities or smart buildings environments.

Regarding the application of the CP-ABE scheme in scenarios where IoT devices are involved, there are some recent proposals that try to accommodate the needs of the scheme to exploit its benefits. The approach proposed in [21] presents a scheme in which IoT devices delegate expensive CP-ABE cryptographic operations in a more powerful device, called assisting node, which is assumed to be trusted. The results of these operations are given to the sensor, which is responsible for computing the ciphertext. The work presented in [22] offers a lightweight KP-ABE scheme to be used in constrained devices. Its main contribution is that the proposed scheme is not based on bilinear pairings, but on the use of *Elliptic Curve Cryptography* (ECC) [23]. Authors in [24] present AGREE, an approach to exploit energy harvesting opportunities to pre-compute and cache suitably chosen CP-ABE-encrypted keys. While their work is focused on WSNs, it could be also considered in the context of IoT devices with tight resource constraints. Moreover, [25] and [26] proposed the integration of the CP-ABE scheme with the *Message Queuing Telemetry Transport* (MQTT) protocol [27], in order to exploit the advantages provided by the publish/subscribe communication pattern. Furthermore, the work proposed by [28] addresses the size of CP-ABE keys, as an aspect limiting its applicability on IoT devices. Towards this end, it proposes a CP-ABE scheme with constant-size secret keys independent of the number of attributes. Additionally, [29] and [30] provide comprehensive performance evaluations of the main CP-ABE operations, by considering different hardware and security levels.

These proposals provide different insights about the application of CP-ABE on IoT-enabled scenarios. However, while these works partially address some of the aspects derived from this integration, the proposed work is intended to provide a more comprehensive and practical approach through the integration with different technologies to realize and implement the concept of *bubble*. Firstly, unlike previous works, the proposed approach is based on our IdM system [8] by considering the inclusion of the System for Cross-domain Identity Management (SCIM) standard [31] that provides a common user schema, extension model and well-defined REST API. In this work, the SCIM standard is used to define users' attributes, so that CP-ABE keys, and consequently the definition of bubbles are based on these values. Secondly, the concept of bubble has been defined by considering the OMA NGSI information model [32] to represent it. OMA NGSI specifications provide a common data model and powerful interfaces for finding information, and it is widely used on current IoT deployments. Thirdly, the resulting approach has been deployed on a real testbed and integrated into the FI-WARE platform, in order to leverage the benefits of a CP-ABE based approach on an already established sharing ecosystem.

It should be noted that, in contrast to [33] where security aspects are enforced by the central platform, our proposal moves the enforcement of security aspects to the end devices, in order to ensure end-to-end data protection. Such approach could be also exploited by considering the main notions from the so-called *Social Internet of Things* (SIoT) [34], so different relationships between devices' owners can drive the way in which data from these devices are protected. These aspects are further explained in the following sections.

3.3 Preliminaries

Before the description of the proposed approach, this section gives an overview of the main concepts and mechanisms that have been used in this work.

3.3.1 SocIoTal bubbles

In order to address scalability requirements in IoT, many current scenarios are based on the massive sharing of data among groups of devices, users and services. These scenarios typically make use of infrastructures that allow communicating entities to remain decoupled. However, the application of security mechanisms is a challenging aspect, specially when the entities involved may be unknown to each other. To address these needs, the SocIoTal project explores the impact of the relationships between people and their smart objects, in order to establish groups or coalitions where information is shared following certain security conditions. In particular, we consider the relationships defined in [34], as a starting point to define security restrictions for a controlled data sharing.

To deal with the dynamism and ephemeral nature of relationships among people and their associated smart objects, SocIoTal envisions the use of two different kinds of groups. On the one hand, a *Community* [33] is defined as a group of users and smart objects that is statically created and managed. Consequently, the user responsible of the bubble has to manage explicitly the joining and leaving actions of other users. On the other hand, a *Bubble* is considered as a group of smart objects whose associated data are shared under a set of security restrictions based on attributes. This way, users are *implicit members* of the bubble as a consequence of satisfying a particular combination of attributes. These bubbles can be created according to preferences that are specified by the owners of such devices, stating different constraints about the relationships that can be established by their smart objects with other entities. As an example, Figure 3.1 shows a high-level representation of a bubble. In this case, Alice and Bob maintain a *Ownership relationship* with their corresponding devices, so each user and his/her devices make up an *Ownership bubble*. Furthermore, Alice and Bob work in the same organization, so they also keep a *Co-work relationship*. As a consequence of this relationship, Alice's car is accessible to Bob, while Bob's video camera is also available to Alice. Then, these devices and their owners state a *Co-work relationship enabled bubble*.

As already mentioned, in order to realize the bubbles' foundations, we propose the use of the CP-ABE scheme. Thus, bubble's security restrictions are defined by using specific combinations or policies of attributes. Therefore, users do not need to be explicitly associated to a bubble; they are implicitly linked because of their attributes. In this sense, only users satisfying such policies will be able to access data from the smart objects of a particular bubble.

3.3.2 Ciphertext-Policy Attribute-Based Encryption

While symmetric key cryptography has been considered as cryptographic solution on emerging IoT scenarios [35], the cumbersome key management and distribution of these approaches hamper a suitable level of scalability to be provided. In contrast, through the use of traditional certificate-based public key cryptography, information can be encrypted to be accessible to a single specific receiver. These limitations require the use of more sophisticated cryptographic approaches in order to provide the bubbles' security functionality previously described.

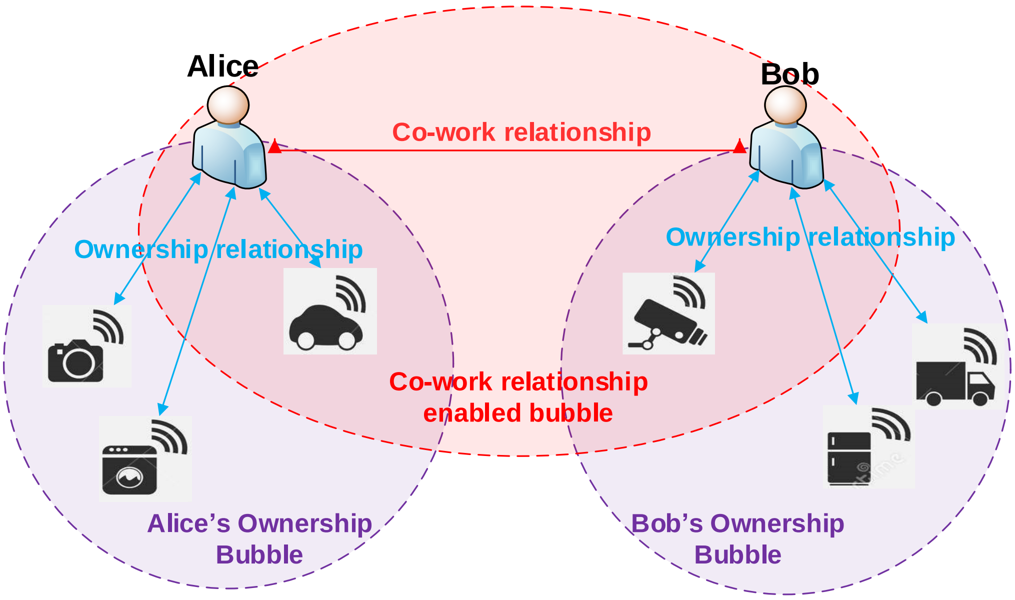


Figure 3.1: Bubble high-level representation

In this sense, *Identity-Based Encryption* (IBE) [36] is an alternative scheme, in which the concept of identity is not determined by a public key, but a string. Therefore, an entity could make a specific information only accessible to other users whose identity is described by a specific string. Based on this, *Attribute-Based Encryption* (ABE) [6] represents the generalization of IBE, in which the identity is not represented by a single string, but by a set of attributes related to their identity. In ABE, a piece of data can be made accessible to a set of entities whose real, probably unknown identity, is based on a certain set of attributes.

By using ABE foundations, two alternative approaches were proposed. In KP-ABE, data are encrypted under a set of attributes, while secret keys are associated with combinations or policies of attributes. In contrast, in CP-ABE, a piece of data is encrypted under a policy of attributes, while keys are associated with sets of attributes. Therefore, CP-ABE could be seen as a more intuitive way to apply the concepts of ABE, in which an entity can exert greater control over how the information is disseminated to other entities. The CP-ABE scheme [7] consists of four main algorithms:

- Setup ($\lambda \rightarrow \{PP, MSK\}$). It takes an implicit security parameter λ as input to generate the public parameters PP that are common to all users of the system (for example, the universe of attributes U), as well as a master secret key MSK, which is used to generate secret keys.
- KeyGeneration ($\{MSK, A\}\rightarrow SK_A$). It takes as input the master key MSK and the set A. The result is a private key SK_A.
- Encrypt ($\{PP, M, PT\}\rightarrow CT$). It takes the message M, public parameters PP and a decryption policy PT representing subsets of attributes that are allowed to decrypt M. The result of this algorithm is a ciphertext CT containing PT.
- Decrypt ($\{PP, CT, SK_A\}\rightarrow M$). The decryption algorithm takes as input the public parameters PP, the ciphertext CT with a PT associated, and a private key SK_A. If the set A satisfies the policy PT, the corresponding entity will be able to decrypt CT with SK_A.

These algorithms are employed to realize the security features of the bubble concept. Section 3.5 provides a detailed description about the use of CP-ABE within the proposed approach.

3.3.3 FI-WARE

FI-WARE[6] is a European middleware platform that promotes the development and global deployment of applications for the Future Internet. FI-WARE delivers a reference architecture, as well as the specification and implementation of different open interfaces called *Generic Enablers* (GEs). Among the FI-WARE GEs, the *Identity Management GE* relies on standard protocols, such as the *Security Assertion Markup Language* (SAML) [37] and OAuth [38], to provide authentication and authorization features, which allows to manage users' access to networks, services and applications. The IdM GE is also responsible for the user profile management, as well as Single Sign-On (SSO)and identity federation across different service domains. Keyrock IdM[7] is an open source implementation of the FI-WARE's IdM GE, which, in turn, relies on Keystone[8], providing further functionality to support the SCIM standard. SCIM is intended to reduce the cost and complexity of user management operations through a common user schema, extension model and REST API with a rich, but simple set of operations. Keyrock has been leveraged in the scope of SocIoTal project to enable the management of users and smart objects' identities, so that security credentials can be associated with the IoT identity attributes defined within Keyrock. Indeed, as will be described, the CP-ABE keys employed within the proposed approach are generated according to these attributes, which in turn, are employed to define the security restrictions for a bubble. For further information about the privacy-preserving IdM system for IoT used in our proposal, the reader is referred to [8].

In addition, the FiWare *Publish/Subscribe Context Broker GE* allows publication of context information by entities, referred as *Context Producers*, so that published context information becomes available to other entities, referred as *Context Consumers*, which are interested in processing the published context information. The Orion Context Broker[9] is an implementation of this GE, providing the OMA NGSI-9 and NGSI-10 interfaces [9] to access to the broker. The Context Manager component of SocIoTal architecture [39] is a fork of the Orion Context Broker, which has been extended with new functionality such as context management and access control. Furthermore, as described in the following section, the definition of bubbles follows the OMA NGSI Context Management Information Model [40], in order to provide an interoperable approach with existing solutions and deployments.

Based on these three main building blocks, following sections provide a detailed description of the proposed approach.

3.4 Bubbles scenario architecture and overall functionality

The bubble is the core concept to enable a dynamic and secure data sharing among different users. As already mentioned, a bubble represents a group of devices whose associated data are shared under specific attributes. Before starting the sharing information process within a bubble, it must be modeled and registered so other devices can be added and shared with other users. For this purpose, bubbles are modelled by following the OMA NGSI Context Information Model [40], so they are represented as a *Context Entity* that is intended to group several devices, which are modeled in the same way.

Listing 3.1: Bubble definition

```
{
        ``entities'': [{
                ``id'': ``bubbleID'',
                ``type'': ``http://sociotal.namespace.bubble'',
                ``isPattern'': ``false''
        }],
        ``attributes'': [{
                ``name'': ``BubbleEntity'',
                ``value'': [``bubbleEntityID''],
```

[6] https://catalogue.fiware.org

[7] https://catalogue.fiware.org/enablers/identity-management-keyrock

[8] https://docs.openstack.org/developer/keystone/

[9] https://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker

```
10                  ‘‘type’’: ‘‘http://sociotal.namespace.bubble.bubbleEntity’’
11          },
12          {
13                  ‘‘name’’: ‘‘attributeName’’,
14                  ‘‘value’’: ‘‘attributeValue’’,
15                  ‘‘type’’: ‘‘attributeType’’
16                  ‘‘metadata’’: [{
17                          ‘‘name’’: ‘‘BubbleRequired’’,
18                          ‘‘value’’: ‘‘true’’,
19                          ‘‘type’’: ‘‘http://sociotal.namespace.bubble.required’’
20                  }]
21          }
22          .
23          .
24          .
25          ]
26  }
```

Based on this, Listing 3.1 shows the main scheme to define a bubble:

- **entities**: the entities (bubbles) to be registered. It includes the bubble's id, the type (*http://sociotal.namespace.bubble*) and the *isPattern* field (lines 3-5), which is always stated to *false* when registering a new bubble.
- **attributes**: it makes reference to a list of attributes related to the bubble(s) to be registered by specifying their name, type and value. In the Listing 3.1, the attribute *BubbleEntity*, type *http://sociotal.namespace.bubbl e.bubbleEntity*, is providing the list of entities' id that are part of this bubble (lines 8-10). In addition to specify the list of entities that are composing the bubble, it is necessary to define which attributes are required to get access to the information provided by the bubble's entities (lines 13-21). For this purpose, such attributes include a set of metadata (lines 16-20) that indicate the *http://sociotal.namespace.bubble.required* type. For these attributes, by following the SCIM schema, the list of supported types is:
  - urn:ietf:params:scim:schemas:core:2.0:organization
  - urn:ietf:params:scim:schemas:core:2.0:department
  - urn:ietf:params:scim:schemas:core:2.0:streetAddress
  - urn:ietf:params:scim:schemas:core:2.0:locality
  - urn:ietf:params:scim:schemas:core:2.0:postalCode
  - urn:ietf:params:scim:schemas:core:2.0:country.

Furthermore, it should be noted that other attributes can be employed to restrict how information is to be shared within a bubble.

Based on the previous definition of *bubble* and the approaches described in Section 3.3, Figure 3.2 shows an overview of the proposed scenario. In particular, as already mentioned, the CP-ABE cryptographic scheme has been employed to manage the security aspects associated to the creation and management of bubbles. Furthermore, in the scope of the SocIoTal project, some of the FIWARE GEs have been integrated and extended to realize the intended functionality.

For the sake of clarity, we adopt a producer/consumer approach in which a user or device can act as a data *producer* or *consumer* at any time [41]. Furthermore, the *SocIoTal Platform* is intended to embrace different infrastructure components to cope with key management and data sharing procedures. In particular, the main identified components are:

- **Web User Environment (WUE)**: it represents the central point where users define their bubbles through a user-friendly interface.
- **IdM Service**: it acts as a repository of users that are previously registered. In particular, it integrates the FI-WARE Keyrock service, which has been extended to support additional attributes based on the SCIM standard following our previous work [8].
- **CP-ABE Key Manager**: it is responsible for generating the cryptographic material associated to the CP-ABE scheme. Specifically, it generates CP-ABE keys associated to the attributes from the IdM Service

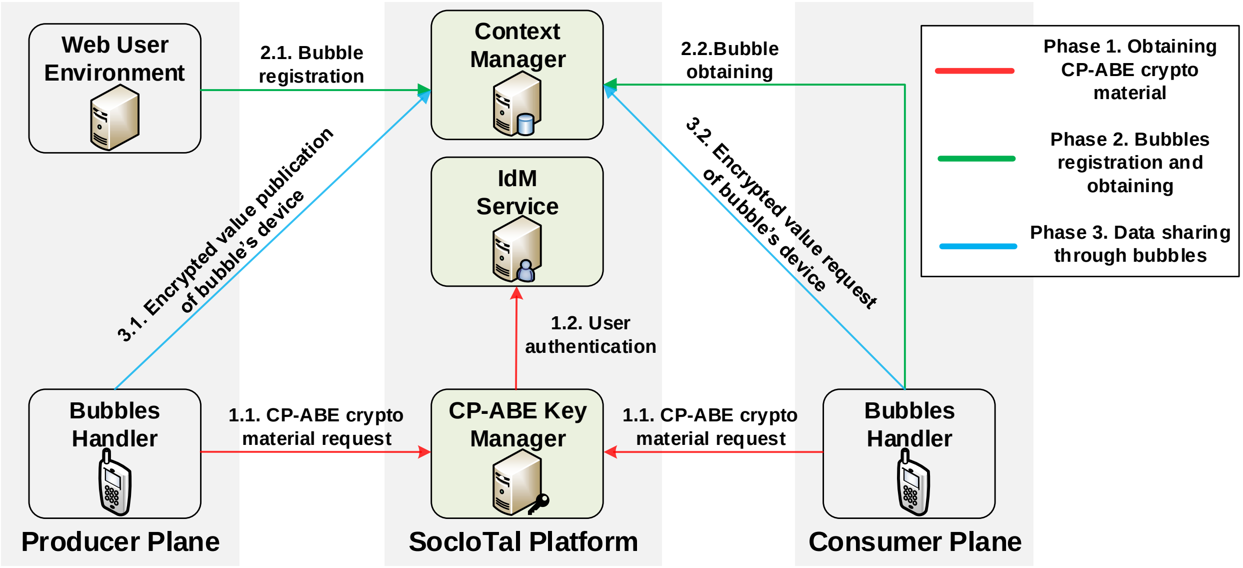

Web User Environment
2.1. Bubble registration
Context Manager
2.2.Bubble obtaining
Phase 1. Obtaining CP-ABE crypto material
Phase 2. Bubbles registration and obtaining
Phase 3. Data sharing through bubbles
IdM Service
3.1. Encrypted value publication of bubble's device
3.2. Encrypted value request of bubble's device
1.2. User authentication
Bubbles Handler
1.1. CP-ABE crypto material request
CP-ABE Key Manager
1.1. CP-ABE crypto material request
Bubbles Handler
Producer Plane
SocIoTal Platform
Consumer Plane

Figure 3.2: SocIoTal bubbles registration, discovery and operation overview

for a particular user.

- **Context Manager**: it is the central point where devices and bubbles are registered to enable a secure sharing of devices' data among different users.
- **Bubbles Handler**: it represents a user application to get registered bubbles and to share devices' data by using CP-ABE encryption and decryption algorithms.

In addition to the main components to realize the bubbles approach, previous figure shows an overview of the main interactions among such components for that. Firstly, during the *Phase 1. Obtaining CP-ABE crypto material*, a user employs her *Bubbles Handler* to request a CP-ABE key and the public parameters that are required to encryption and decryption algorithms (step 1.1). Then, the *CP-ABE Key Manager* authenticates the user through the *IdM Service* to get the attributes associated to her (step 1.2). These attributes are used to generate the corresponding CP-ABE key for that user that is delivered together the public parameters. It should be pointed out that this process is only performed once (or until the cryptographic material is revoked), and it is independent to the definition of different bubbles, so key management aspects are simplified. Indeed, changing a bubble's policy does not require the revocation or generation of new cryptographic material.

During the *Phase 2. Bubbles registration and obtaining*, a user employs the *WUE* to register bubbles (step 2.1) by specifying a set of devices to be added and the attributes that are required to share within that bubble. This bubble is registered in the *Context Manager* by following the data model previously described. Then, other users, through the *Bubbles Handler*, try to discover bubbles in the Context Manager (step 2.2) by using NGSI9-10 interfaces (e.g. via *queryContext* method).

After previous processes have been completed, in the *Phase 3. Data sharing through bubbles*, a user updates the value of an entity of such bubble, by making use of CP-ABE indicating the policy that is employed to encrypt such value. This policy makes reference to the attributes that are specified in the bubble definition. For example, a CP-ABE policy could be represented as "organization=umu AND department=diic". The entity is updated with the encrypted value in the Context Manager by using NGSI-9/10 interfaces via the *updateContext* method (step 3.1). Finally, a user can get the new encrypted value from the Context Manager through the *queryContext* operation; in this case, only those users who satisfy the CP-ABE policy associated to that bubble will be able to decrypt the updated value (step 3.2).

3.5 Bubbles scenario interactions

After providing a general description of the proposed scenario, this section is intended to delve into the 3 main phases previously defined. Figure 3.3 provides a more detailed overview of the required interactions for each phase that are explained below.

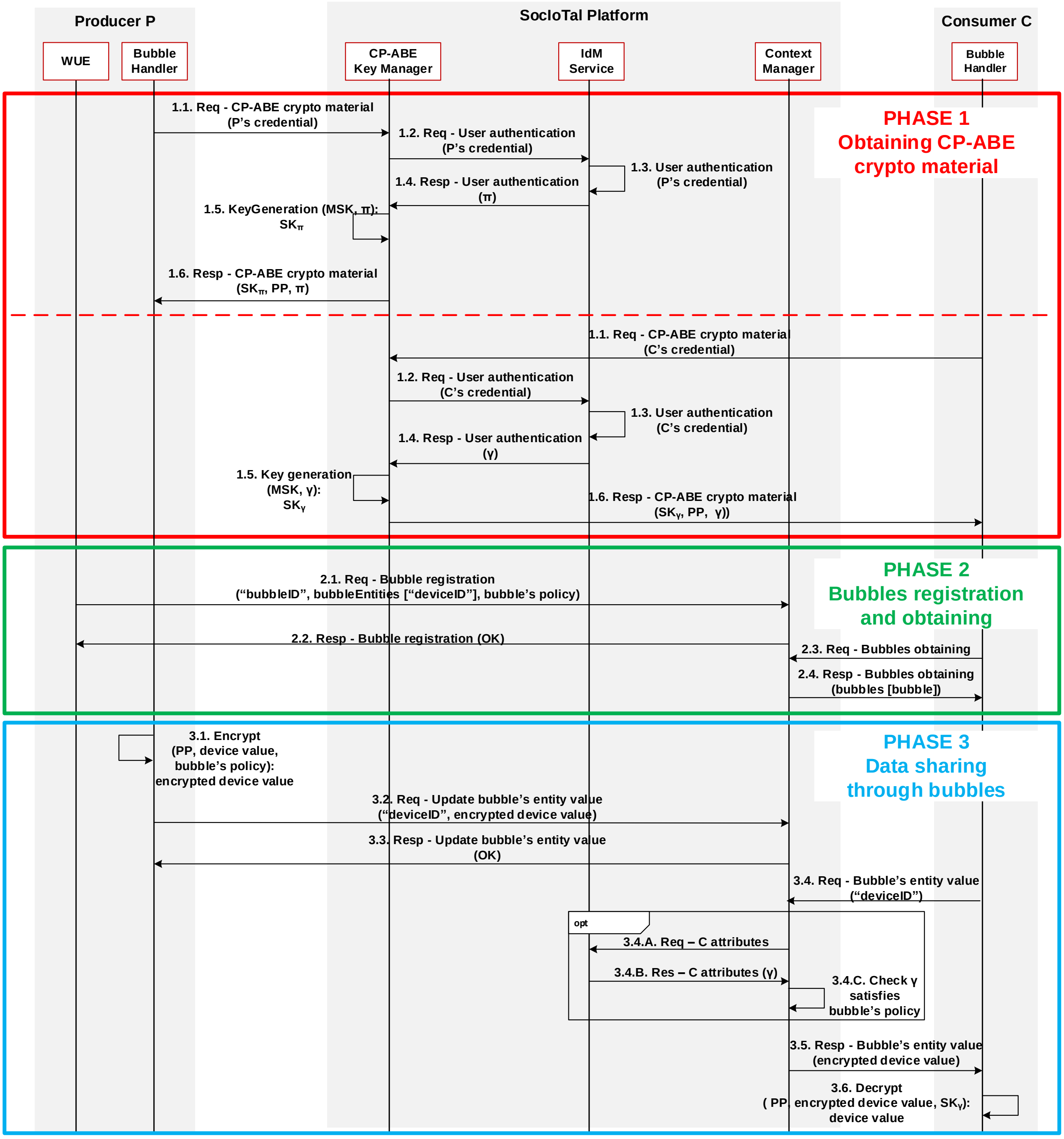

Producer P
SocIoTal Platform
Consumer C
WUE
Bubble Handler
CP-ABE Key Manager
IdM Service
Context Manager
Bubble Handler
PHASE 1 Obtaining CP-ABE crypto material
1.1. Req - CP-ABE crypto material (P's credential)
1.2. Req - User authentication (P's credential)
1.3. User authentication (P's credential)
1.4. Resp - User authentication (π)
1.5. KeyGeneration (MSK, π): SKπ
1.6. Resp - CP-ABE crypto material (SKπ, PP, π)
1.1. Req - CP-ABE crypto material (C's credential)
1.2. Req - User authentication (C's credential)
1.3. User authentication (C's credential)
1.4. Resp - User authentication (γ)
1.5. Key generation (MSK, γ): SKγ
1.6. Resp - CP-ABE crypto material (SKγ, PP, γ))
PHASE 2 Bubbles registration and obtaining
2.1. Req - Bubble registration ("bubbleID", bubbleEntities ["deviceID"], bubble's policy)
2.2. Resp - Bubble registration (OK)
2.3. Req - Bubbles obtaining
2.4. Resp - Bubbles obtaining (bubbles [bubble])
PHASE 3 Data sharing through bubbles
3.1. Encrypt (PP, device value, bubble's policy): encrypted device value
3.2. Req - Update bubble's entity value ("deviceID", encrypted device value)
3.3. Resp - Update bubble's entity value (OK)
3.4. Req - Bubble's entity value ("deviceID")
opt
3.4.A. Req – C attributes
3.4.B. Res – C attributes (γ)
3.4.C. Check γ satisfies bubble's policy
3.5. Resp - Bubble's entity value (encrypted device value)
3.6. Decrypt (PP, encrypted device value, SKγ): device value

Figure 3.3: SocIoTal bubbles registration, discovery and operation overview

3.5.1 Obtaining CP-ABE crypto material

As described in the previous section, initially the users try to obtain the required cryptographic material to protect their data within a bubble. For this purpose, while it is not shown in the figure, two previous processes are assumed. Firstly, the CP-ABE Key Manager has already previously run the *Setup* algorithm (see Section 3.3.2), so the public parameters *PP* and the master key *MSK* have already been generated. It should be noted that this operation is only required initially to setup the CP-ABE ecosystem. Secondly, users have already been registered in the *IdM Service*; therefore, this service has a set of attributes for each user. For registration purposes, a user can employ the *WUE* as described in [33].

Following the previous notation and Figure 3.3, we consider two users acting as a data *Producer P* and a data *Consumer C*. It should be noted the steps associated to the first phase are repeated by each entity. According to it, a user makes usage of the *Bubble Handler* that sends a request to the endpoint where the *CP-ABE Key Manager* accepts requests for generating cryptographic material. This request includes an authentication credential (P's/C's credential) (step 1.1) to ensure that only legitimate users will be able to obtain the appropriate keys. Then, the CP-ABE Key Manager queries the *IdM Service* in order to authenticate the requesting user (step 1.2). As already mentioned, the IdM Service integrates an extended version of the FIWARE Keyrock IdM that includes support for additional attributes following the SCIM standard. Furthermore, this service supports different authentication mechanisms (step 1.3) including login/password and certificate-based procedures, so that the credential could be instantiated by considering different approaches. It should be noted that this authentication process could be also performed in a privacy-preserving way based on the use of Idemix [42] by following our previous proposal in [8]. In case of a successful authentication process, the IdM Service returns the set of user attributes (π/γ) to the CP-ABE Key Manager (step 1.4). Then, this entity runs the *KeyGeneration* algorithm (see Section 3.3.2) to generate a new CP-ABE key that is associated to those attributes. Consequently, the CP-ABE keys for P and C are generated as (step 1.5):

$$KeyGeneration\ (\{MSK, \pi\} \rightarrow SK_{\pi})$$
$$KeyGeneration\ (\{MSK, \gamma\} \rightarrow SK_{\gamma})$$

Then, each CP-ABE key is sent together with the public parameters PP in the response to the requesting user (step 1.6). For this, both elements are base64-encoded and included in a JSON object following the example in Listing 3.3. In addition to the CP-ABE key and public parameters, the set of attributes associated to the key is also included in the response, so that users can easily visualize their own attributes.

Listing 3.2: CP-ABE Key Manager response example

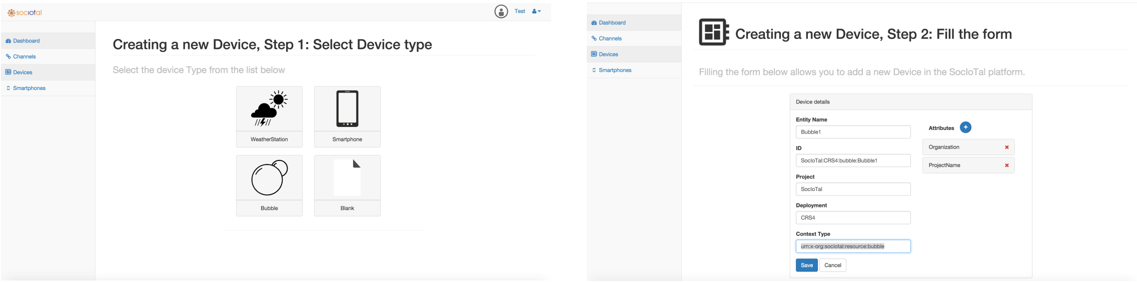
```
{
``cpabekey'': ``xNlksSd_13ndwe9...''
``publicparam'': ``zcsDko32urnlAdanx...''
``attributes'': [
        {
        ``name'': ``organization'',
        ``value'': ``umu'',
        ``type'': ``urn:ietf:params:scim:schemas:core:2.0:organization''
        },
        .
        .
        .
        ]
}
```

As mentioned above, this initial phase is only performed when a new key or cryptographic parameters are required (for example, because they have expired). It should be noted that unlike symmetric cryptography based approaches, only public parameters need to be shared by all users. In this sense, the definition or elimination of a bubble does not require the generation or revocation of any cryptographic material, so the aspects of key management are significantly reduced.

It should be noted that this initial phase is also required in case the cryptographic material is revoked. According to our scenario, CP-ABE keys are associated to the attributes stored in the IdM Service. Therefore, when a user's attribute (for example, her organization) changes, the associated CP-ABE key is not valid anymore. Although the revocation aspects do not represent the focus of our work, different approaches can be considered to address this issue. On the one hand, the use of techniques based on proxy re-encryption [43] has been traditionally used for this purpose, but it requires the definition of new components in the architecture and an expensive process to update the key of each participant. On the other hand, the use of expiration time attributes can help to mitigate the revocation issue, as suggested by [7]. Indeed, in our previous work, the use of expiration time is associated with symmetric keys that are distributed by using CP-ABE [44]. However, in addition to practical considerations about which amount of time is suitable, it does not address the problem associated to an explicit change of a user's attribute. In our case, considering the proposed architecture, a potential approach is the use of the IdM Service to check the current attributes of a user. In this way, each time a user queries the Context Manager, this entity checks the attributes of the requesting user, as will be further described in Section 3.5.3.

3.5.2 Bubbles registration and obtaining

When a user has already completed the previous phase, she can use the WUE for the creation and definition of a bubble, associating a set of devices and the required attributes to access devices' data. To do this, the WUE provides a friendly user interface so that a user is able to create a bubble in the same way she registers other devices (e.g. her weather station or smartphone).



Creating a new Device, Step 1: Select Device type

Select the device Type from the list below

Creating a new Device, Step 2: Fill the form

Filling the form below allows you to add a new Device in the SocIoTal platform.

Device details

Entity Name: Bubble1

ID: SocIoTal:CRS4:bubble:Bubble1

Project: SocIoTal

Deployment: CRS4

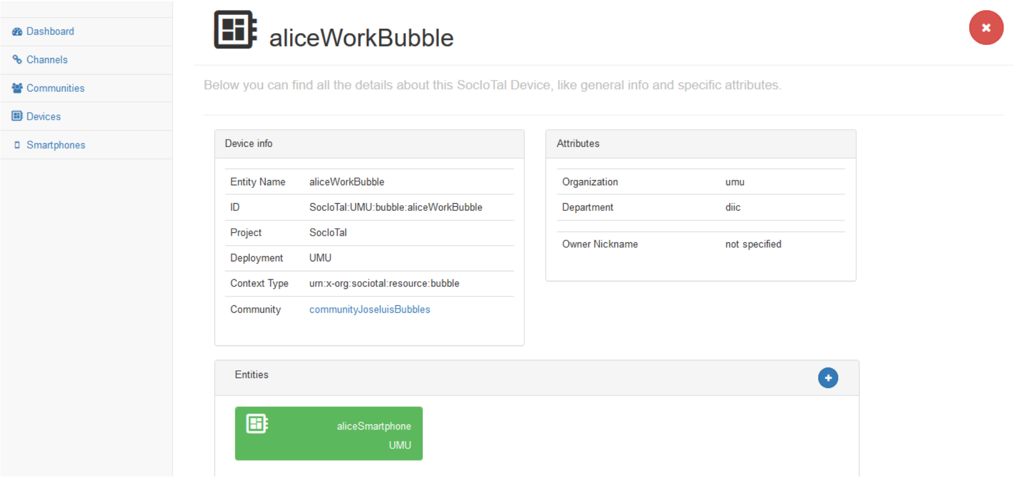
Context Type: urn:x-org:sociotal:resource:bubble

Attributes: Organization, ProjectName

(**a**) SocIoTal bubbles creation (**b**) SocIoTal bubbles definition

Figure 3.4: WUE examples for creating and defining a bubble

Figure 3.4 shows an example of WUE screenshots in which a user is enabled to create and define a bubble. As shown in Figure 3.4 (b), the user can define different details for the bubble (e.g., name, ID or type), as well as the attributes that will be required to get access to the devices data included in the bubble.



aliceWorkBubble

Below you can find all the details about this SocIoTal Device, like general info and specific attributes.

Device info

Entity Name	aliceWorkBubble
ID	SocIoTal:UMU:bubble:aliceWorkBubble
Project	SocIoTal
Deployment	UMU
Context Type	urn:x-org:sociotal:resource:bubble
Community	communityJoseluisBubbles

Attributes

Organization	umu
Department	diic
Owner Nickname	not specified

Entities

aliceSmartphone
UMU

Figure 3.5: Bubble definition example view at the WUE

After creating a bubble, the user can associate her devices (or some of them) to it. Figure 3.5 shows an example of a bubble created by the user “Alice” (acting as the *Producer P*) with the name “aliceWorkBubble”. In this case, it is assumed that she has previously registered her smartphone with the name “aliceSmartphone” by using the WUE (see Figure 3.4 (a)). In addition to the bubble details, such as ID or type, Alice has associated her smartphone with the new bubble. She has also specified the *Organization=umu* and *Department=diic* attributes as the attributes required to access the information generated by devices associated to such bubble. Consequently, all the information provided by the *AliceSmartphone* entity will be accessible only for users who satisfy those attributes.

Following this example, Listing 3.3 shows the bubble definition that is included as the payload of a NGSI-9 *registerContext* (or a NGSI-10 *updateContext*). This operation is represented by the step 2.1 in Figure 3.3, where *bubbleID* is “aliceWorkBubble”, the list of *bubbleEntities* is only composed by “AliceSmartphone”, and the *bubble’s policy* is represented by the conjunction of attributes that are marked with type “http://sociotal.namespace.bubble.required” (i.e., *organization=umu AND department=DIIC*)

Listing 3.3: Bubble registration example

```
‘‘contextRegistrations’’: [{
        ‘‘entities’’: [{
                ‘‘id’’: ‘‘AliceWorkBubble’’,
                ‘‘type’’: ‘‘http://sociotal.namespace.bubble’’,
                ‘‘isPattern’’: ‘‘false’’
        }],
        ‘‘attributes’’: [{
                ‘‘name’’: ‘‘BubbleEntity’’,
                ‘‘value’’: [‘‘AliceSmartphone’’],
                ‘‘type’’: ‘‘http://sociotal.namespace.bubble.bubbleEntity’’
        },
        {
                ‘‘name’’: ‘‘organization’’,
                ‘‘value’’: ‘‘umu’’,
                ‘‘type’’: ‘‘urn:ietf:params:scim:schemas:core:2.0:organization’’,
                ‘‘metadata’’: [{
                        ‘‘name’’: ‘‘BubbleRequired’’,
                        ‘‘value’’: ‘‘true’’,
                        ‘‘type’’: ‘‘http://sociotal.namespace.bubble.required’’
                }]
        },
        {
                ‘‘name’’: ‘‘department’’,
                ‘‘value’’: ‘‘diic’’,
                ‘‘type’’: ‘‘urn:ietf:params:scim:schemas:core:2.0:department’’,
                ‘‘metadata’’: [{
                        ‘‘name’’: ‘‘BubbleRequired’’,
                        ‘‘value’’: ‘‘true’’,
                        ‘‘type’’: ‘‘http://sociotal.namespace.bubble.required’’
                }]
        }]
}
```

In this case, based on Section 3.3.1, we assume that Alice has a *Ownership relationship* with “aliceSmartphone”, since she is her owner. The definition of the attributes “Organization=umu” and “Department=diic” can be seen as the specification of a *Co-work relationship*, so that users working within the *diic* department at *umu* will be able to access the data generated by Alice’s smartphone.

Listing 3.4: Bubbles query message example

```
{
  ‘‘entities’’: [
    {
      ‘‘type’’: ‘‘urn:x–org:sociotal:resource:bubble’’,
      ‘‘isPattern’’: ‘‘true’’,
      ‘‘id’’: ‘‘*’’
    }
  ],
  ‘‘attributes’’: []
}
```

When the bubble is registered, the Context Manager sends a message indicating the process was successfully performed (step 2.2). Then, a user acting as the *Consumer C*, accesses the Context Manager in order to obtain information about the registered bubbles. Listing 3.4 shows an example of the payload contained within a OMA NGSI-10 *queryContext* method to get all entities of type “urn:x-org:sociotal:resource:bubble” (step 2.3).

Listing 3.5: Bubbles response message example

```
{``contextResponses'': [
  {
                      ``statusCode'': {
                      ``reasonPhrase'': ``OK'',
                      ``code'': ``200''
    },
    ``contextElement'': {
      ``id'': ``SocIoTal:UMU:bubble:aliceWorkBubble'',
      ``attributes'': [
        {
          ``name'': ``Organization'',
          ``value'': ``umu'',
          ``type'': ``http://sensorml.com/ont/swe/property/OrganizationName'',
          ``metadata'': [
            {
              ...
            }
          ]
        },
        {
          ``name'': ``Department'',
          ``value'': ``diic'',
          ``type'': ``urn:ietf:params:scim:schemas:core:2.0:department'',
          ``metadata'': [
            {
              ...
            }
          ]
        },
        {
          ``name'': ``owner'',
          ``value'': ``Alice'',
          ``type'': ``identitier:UUID''
        },
        {
          ``name'': ``BubbleEntity'',
          ``value'': [
            ``SocIoTal:UMU:smartphone:aliceSmartphone''
          ],
          ``type'': ``http://sociotal.namespace.bubble.bubbleEntity''
        }
      ],
      ``type'': ``urn:x-org:sociotal:resource:bubble'',
      ``isPattern'': ``false''
    }
  }
]
```

Following the example, Listing 3.5 shows the response to the previous request in which the bubble information “aliceWorkBubble” is obtained (step 2.4). In this way, a user is able to visualize the bubbles registered in the Context Manager, as well as the attributes required to be able to access the data of the associated devices.

3.5.3 Data sharing through bubbles

Once a user has registered a bubble through the WUE, she can use her Bubble Handler to share protected data by using the Context Manager. In this way, following the previous example, Alice (acting as the *Producer P*) decides to protect the location information that is detected by her smartphone. To do this, the Bubble Handler obtains information about the bubbles registered by Alice in the Context Manager. Since the device “aliceSmartphone” is associated with the bubble “aliceWorkBubble”, the Bubble Handler detects that the device’s data must be protected by using the bubble’s policy (“Organization = umu” AND “Department = diic”). It should be noted that this functionality is not reflected in the sequence diagram, since the Bubble Handler will be usually stored this information, so the interaction with the Context Manager is not required. The Bubble Handler runs the *Encrypt* algorithm (see Section 3.3.2) to encrypt the location data by using the mentioned policy as (step 3.1):

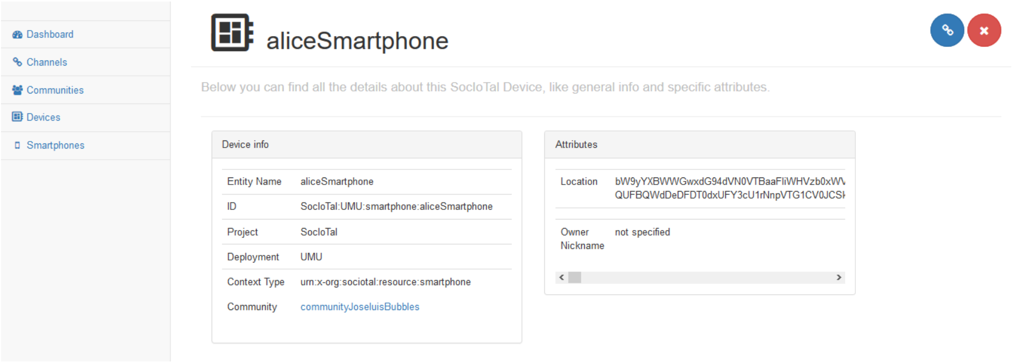
$$\text{Encrypt}\ (\{PP, devicevalue, (\text{“}Organization = umu\text{”}$$
$$AND\text{“}Department = diic''\text{)}\}\rightarrow encrypteddevicevalue)$$

where device value is represented by the location data, following the example. Then, the resulting ciphertext *encrypted device value* is then included within an NGSI-10 *updateContext* message, as shown in the Listing 3.6 that is sent to the Context Manager (step 3.2). This message is in turn answered by the Context Manager indicating the value was successfully updated (step 3.3).

Listing 3.6: Context entity update message example

```
{``contextElements'': [
    {
      ``type'': ``urn:x–org:sociotal:resource:smartphone'',
      ``isPattern'': ``false'',
      ``id'': ``SocIoTal:UMU:smartphone:aliceSmartphone'',
      ``attributes'': [
        {
          ``name'': ``Location'',
          (*\bfseries ``value'': ``TzVPV3NCSlVONlVwMHErSUZxbC9PUT09...''*),
          ``type'': ``http://sensorml.com/ont/swe/property/Location'',
          ``metadatas'': [
            {
              ...
            }
          ]
        }
      ]
    }
  ],
  ``updateAction'': ``UPDATE''
}
```

At this point, it should be noted that even though the information is shared through the Context Manager, this entity does not have access to the information provided by the different entities, since data are encrypted on an end-to-end basis. In fact, Figure 3.6 shows the encrypted value corresponding the smartphone's location when it is queried through the WUE.



aliceSmartphone

Below you can find all the details about this SocIoTal Device, like general info and specific attributes.

Device info

Entity Name	aliceSmartphone
ID	SocIoTal:UMU:smartphone:aliceSmartphone
Project	SocIoTal
Deployment	UMU
Context Type	urn:x-org:sociotal:resource:smartphone
Community	communityJoseluisBubbles

Attributes

Location	bW9yYXBWWGwxdG94dVN0VTBaaFliWHVzb0xWV QUFBQWdDeDFDT0dxUFY3cU1rNnpVTG1CV0JCSk
Owner Nickname	not specified

Figure 3.6: Alice's smartphone view at the WUE when location data is encrypted

After that, a user (acting as a *Consumer C*) tries to access Alice's position. From the message obtained in the previous phase (see Listing 3.5), she knows that the "aliceSmartphone" device is associated with the "aliceWorkBubble" bubble. Therefore, she can use his Bubble Handler to obtain the data associated to it by using the NGSI-10 *queryContext* method with the content shown in Listing 3.7 (step 3.4).

Listing 3.7: Device's encrypted data query message example

```
{
  ``entities'': [
    {
      ``type'': ``urn:x–org:sociotal:resource:smartphone'',
      ``isPattern'': ``false'',
      ``id'': ``SocIoTal:UMU:smartphone:aliceSmartphone''
    }
  ],
  ``attributes'': []
}
```

Considering our architecture, as already mentioned in Section 3.5.1, we have added optional steps to deal with issues associated to attributes delegation and revocation. In particular, when the Context Manager receives the previous message, it queries the IdM service to get the current attributes of the requesting user (step 3.4.A). Then, the set of attributes (γ) is sent back to the Context Manager (step 3.4.B). Since the policy of a bubble is still accessible to the Context Manager, it can check that the set of attributes satisfy the bubble's policy (step 3.4.C). Accordingly, if this condition is not satisfied, the Context Manager does not provide the requested data. Instead, it returns an error message, so that the requesting user is redirected to the CP-ABE Key Manager, in order to update her CP-ABE key, so that the *Phase 1* is carried out again. Otherwise, if the verification process is successful, the Context Manager returns the message in Listing 3.8 in which the location information of the device is encrypted (step 3.5). In fact, the location value corresponds to the output of the CP-ABE encryption that is encoded in base64. Once the above message is obtained, the *Consumer C* uses her Bubble Handler to execute the CP-ABE Decrypt algorithm, (see Section 3.3.2) by using her private key and parameters obtained in the first phase, as well as the encrypted location value, as follows (step 3.6).

$$\text{Decrypt}\ (\{PP,\ encrypteddevicevalue,\ SK_{\gamma}\}{\rightarrow}devicevalue)$$

This way, in case the user's CP-ABE key satisfies the bubble's policy (*"Organization = umu" AND "Department = diic"*) she will be able to get the value of *locationData* (i.e., *device value*)

Listing 3.8: Encrypted entity response example

```
{``contextResponses'': [
    {
      ``statusCode'': {
        ``reasonPhrase'': ``OK'',
        ``code'': ``200''
      },
      ``contextElement'': {
        ``id'': ``SocIoTal:UMU:smartphone:aliceSmartphone'',
        ``attributes'': [
          {
            ``name'': ``Location'',
           (*\bfseries ``value'': ``TzVPV3NCSlVONlVwMHErSUZxbC9PUT09...''*),
            ``type'': ``http://sensorml.com/ont/swe/property/Location'',
            ``metadatas'': [
              {
                ...
              }
            ]
          },
          {
            ``name'': ``owner'',
            ``value'': ``Alice'',
            ``type'': ``identitier:UUID''
          }
        ],
        ``type'': ``urn:x-org:sociotal:resource:smartphone'',
        ``isPattern'': ``false''
      }
    }
  ]
}
```

It should be noted that, even considering the additional interactions to cope with delegation/revocation aspects, two main advantages of the use of CP-ABE are still leveraged. Firstly, while the Context Manager is able to access the information related to a bubble (including the associated policy), it cannot access the data that is shared between users through such bubbles. Secondly, the potential problems associated with an uncontrolled delegation of attributes are also mitigated due to the verification of attributes for each request to the Context Manager. In addition, as already mentioned, changes in the policy of a bubble do not require the revocation or update of the cryptographic material. While it is not the focus of the paper, part of our future work is related to the integration of more efficient revocation schemes [45] that could be potentially analyzed in the context of IoT environments.

3.6 Use case: sharing location data through bubbles in indoor environments

Following with the example of sharing location data, in this section we describe a scenario in which a user's location data are shared through the proposed approach in a smart building. Indeed, according to the *European Alliance of Companies for Energy Efficiency in Buildings* (EuroACE [46]), we spend over 90% of our time in buildings. Consequently, we consider the uncontrolled disclosure of people's location data in such environments can facilitate tracking tasks that can potentially harm their privacy through the inference of their daily habits. Furthermore, it should be noted that typical GPS-based approaches are not practical in these scenarios due to the lack of signal.

Towards this end, the proposed localization system is intended to be used in indoor environments (e.g. a smart building) and it is based on the use of two different low-level devices. The first ones act as *anchors*, which are static and calibrated. The second ones are mobile low-level devices worn by people. They act as *sinks* of *Received Signal Strength Indicators* (RSSI) measured at each receiving of a beacon message sent by the anchors. Furthermore, they are connected to the Context Manager via a local gateway through a wireless communication channel. These sinks are able to acquire raw time-stamped location-dependent information (either absolute information or relative information with respect to their neighbors) that can be interpreted further by any localization engine (i.e., in terms of ranging, positioning or position tracking) embedded in the gateway. Thanks to dedicated algorithms, the coordinates of the moving sink nodes are estimated and subsequently transmitted through OMA NSGI-9/10 to the Context Manager where they are stored as contextual information in the broker database. In this scenario, the WSN nodes are IEEE 802.15.4-enabled devices using OpenMote hardware. The embedded application is Contiki-based. The user position is derived from its distance to anchor nodes. Such distance is computed by using RSSI. The user carries a special device called the mobile node, along with a smartphone for graphical feedback. The goal is to display the real-time user's location on a map on the smartphone.

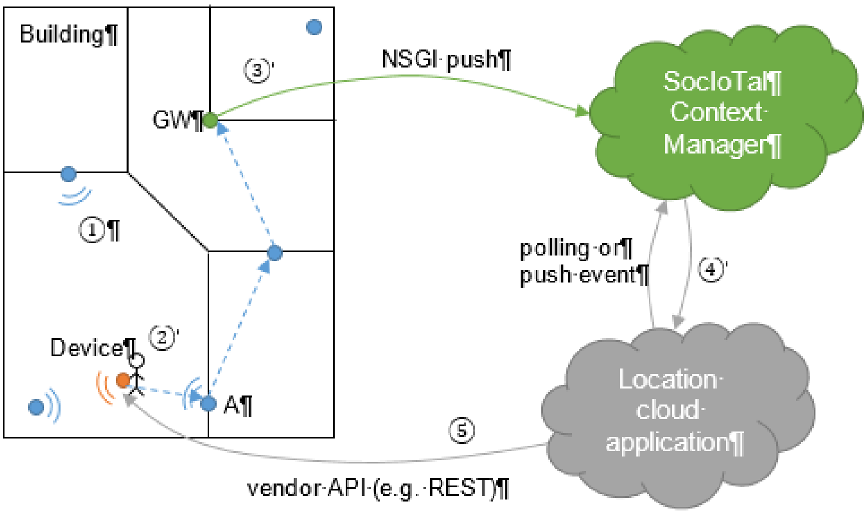


Figure 3.7: The radio localization scenario

The location process follows these steps:

1. **Collection of the RSSI and computation of the relative distances**. The mobile device broadcasts a beacon packet to its in-range neighbors (anchor nodes represented by blue discs with a signal symbol). The neighbors acknowledge the beacon with a reply packet. The mobile node stores the various RSSIs as they arrive from neighbors.
2. **Deduction of the 2D coordinates**. When all RSSIs are received, they are encapsulated in a message sent to the gateway (GW, green disc) using the existing WSN topology; e.g. in Figure 3.7, the route to the gateway starts with node A. The gateway proceeds to compute the estimated location of the device based on the collected RSSIs, using a specific algorithm.

Furthermore, the computed location is pushed to the Context Manager by using OMA NGSI-9/10 messages. Then, the third-party "Location application", hosted in the cloud, receives the updated location of the user/device, and finally, the graphical front-end running on the user's smartphone updates the position (protocol is application-specific).

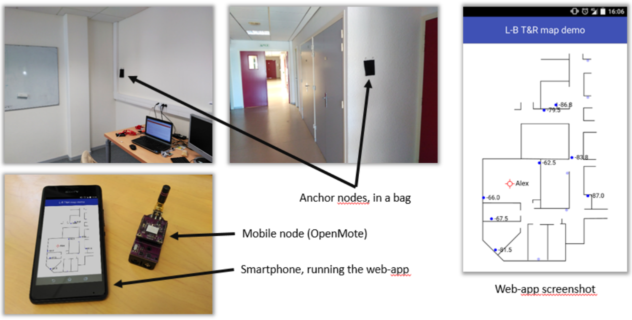


Figure 3.8: Deployment of the radio localization demonstrator in an indoor environment

3.6.1 Collection of the RSSI and computation of the relative distances

The collection of Received Signal Strength Indicator (RSSI) measurements is deployed in the WSN (see Figure 3.8) using the openmote technology. Based on a meshed network topology, the RSSI collected by the mobile sink openmote node are sent to the gateway in order to evaluate the corresponding distance from the sink to the anchors. The radio signal is secured at the MAC layer thanks to the security features provided by the Contiki 3.x embedded OS in the openmotes and the linux 4.4.4 kernel deployed in the local gateway according to the techniques depicted in [47].

3.6.1.0.1 Wireless Connectivity Based on the scenario described, N_a fixed anchors are disseminated at known locations in the environment of interest and so-called "blind" mobile nodes to be positioned. In location-enabled IEEE 802.15.4 networks, the ranging functionality usually relies on received power measurements through the RSSI. In terms of RSSI availability, possible radio links depend at least on the power loss suffered over the transmission range and on the receiver power threshold. It is an intrinsic capability of the hardware device. Accordingly, for a given position occupied by the mobile node, the exploitable 1-hop neighborhood $H = \{i / P_{R_x,i} > S\}$ is thus defined as a set of reachable anchors, if $P_{R_x,i}$ is the received analog power over the link with respect to the i-th anchor and S is the equivalent RSSI detection threshold. The entries $\{I_i = 1 \text{ if } i \in H\}$ of the related connectivity (or adjacency) vector directly account for physically feasible RSSI-enabled links.

3.6.1.0.2 Received Signal Strength Measurements On the receiver side, the analog Received Signal Strength (RSS) is generally expressed as follows:

$$P_{R_x}(d) = P_{R_x,ref} - 10\alpha \log_{10}\left[d/d_{ref}\right] + X_s \tag{3.1}$$

where $P_{R_x,ref}$ is the average power received at the reference distance d_{ref} (generally equal to 1 m), α is the path loss decay exponent and X_s is a random term with standard deviation σ_s. $P_{R_x,ref}$ accounts for the transmit power, transmit and receive antenna gains, and power path loss at the reference distance.

In static scenarios, X_s traditionally accounts for centered Gaussian shadowing. Assuming mobile nodes and/or mobile scatterers, additional small-scale fading (e.g., Rayleigh) is observed around the shadowed received power. However, in most of practical scenarios, these rapid power fluctuations can be significantly mitigated out, and hence neglected, after averaging over several consecutive power measurements so as to draw mostly their deterministic range-dependent component, i.e., considering for ranging $\overline{P_{R_x}}(t) = \frac{1}{K}\sum_{k=1}^{K} P_{R_x}(k,t)$ instead of $P_{R_x}(k,t)$ at time stamp t. Obviously, the size K of the averaging sequence must be judiciously chosen with respect to the channel coherence time (e.g. [48]), given a mobility scenario (typically, pedestrian in our case).

3.6.1.0.3 RSSI-based Range Measurements and Related Errors Based on the previous RSSI measurements collected between pairs of devices, one can determine the corresponding relative distances, given a priori path loss model parameters. Using the median estimator for instance (e.g. [48], [49], [50]), which requires only little prior knowledge about channel model parameters but provides an intermediary level of performance, the range measurement is defined as:

$$\tilde{d} = exp(M) \tag{3.2}$$

where $M = \frac{(P_{R_x,ref} - \overline{P_{R_x}})log(10)}{10\alpha} + log(d_{ref})$.

In [49], the estimation variance affecting $\tilde{d}$ was shown to be:

$$\sigma_d^2 = d^2 exp(s^2)[exp(s^2) - 1] \tag{3.3}$$

where $s = \frac{\sigma_s log(10)}{10\alpha}$.

Optionally, the knowledge of ranging variance can be advantageously incorporated within any subsequent positioning step fed by weighted range measurements. Practically, in real-life systems, this variance can be approximated as $\tilde{\sigma_d}^2$ by incorporating the estimated distance $\tilde{d}$ instead of the actual distance d in the corresponding formulas (e.g. [49]).

Preferably, due to the highly parametric nature of the retained ranging approach, the path loss model parameters must be a priori calibrated in the operating environment and/or, whenever possible, per anchor [50]. This is for instance obtained empirically through standard data fitting, after collecting received power measurements at known distances. In our case, $P_{R_x,ref} = -62$ and $\alpha = 3.5$ have been determined as optimal parameters.

3.6.2 Deduction of the 2D coordinates

3.6.2.0.1 Positioning Algorithm The considered variant for the implemented positioning algorithm relies on a range-based non-linear *Weighted Least Squares* (WLS) optimization procedure. Following a standard centralized synchronous approach [51], the 2D Cartesian coordinates of a given mobile node are determined as the solutions of the following problem:

$$(\hat{x}_m, \hat{y}_m) = \text{argmin}_{(\hat{x},\hat{y})} \sum_{i=1}^{N_a} \frac{I_i}{\sigma_{d,i}^2} \left(\tilde{d}_i - \hat{d}_i\right)^2 \tag{3.4}$$

where $\hat{x}_m$ and $\hat{y}_m$ are the estimated 2D coordinates of the mobile node, $\hat{d}_i = \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2}$ is the estimated distance between the mobile node and the i-th anchor, and I_i represents the force, measurement of the power signal, of the RSSI.

As for optimization solvers, many solutions can be considered, such as Sequential Quadratic Programming (SQP), Linearized Least Squares (LLS) through Taylor Series Expansion (TSE), Gradient Conjugate (GC), Newton-Raphson, or the popular iterative Gradient-Descent (GD) (e.g., [52], [51]), etc. In the following, we consider the latter implementation for simplicity.

Accordingly, given an initial guess $(\hat{x}_m(t,0), \hat{y}_m(t,0))$ at time stamp t and iteration $j = 0$, one can simply compute the WLS solution for any iteration $j > 0$, as follows:

$$\begin{aligned} \hat{x}_m(t,j) &= \hat{x}_m(t,j-1) + \delta_x(t,j), \\ \hat{y}_m(t,j) &= \hat{y}_m(t,j-1) + \delta_y(t,j). \end{aligned} \tag{3.5}$$

where

$$\begin{aligned}&\delta_x(t,j) =\\&\textstyle\sum_{i=1}^{N_a} \beta_i(\tilde{d_i}(t) - \hat{d}_i(t,j-1))\frac{\hat{x}_m(t,j-1)-x_i(t,j-1)}{\hat{d}_i(t,j-1)},\\&\delta_y(t,j) =\\&\textstyle\sum_{i=1}^{N_a} \beta_i(\tilde{d_i}(t) - \hat{d}_i(t,j-1))\frac{\hat{y}_m(t,j-1)-y_i(t,j-1)}{\hat{d}_i(t,j-1)}.\end{aligned} \tag{3.6}$$

with β_i a scaling descent coefficient, for instance proportional to $1/\tilde{\sigma}_{d,i}$ or even more simply equal to 0.1 in our case and $\hat{d}_i(t,j-1)$ the latest estimated distance equal to: $\sqrt{(x_i - \hat{x}_m(t,j-1))^2 + (y_i - \hat{y}_m(t,j-1))^2}$

Different stopping rules can be applied, either based on a maximum tolerated number of iterations (e.g., under constrained computational complexity) or if the norm of the descent progress does not exceed an a priori threshold (e.g., given an a priori spatial resolution to express the solution).

As for initialization, one can use the weighted centroid of all available anchors or alternatively, an arbitrary subset of anchors leading to the strongest RSSI readings.

3.6.3 Integrating the radio localization system into the bubbles approach

In order to describe the integrated scenario by using the bubbles concept, Figure 3.9 shows the sequence diagram with the required steps to share location data by using the proposed localization system. Following with our example, Alice wishes to share her location data obtained from the proposed system to those users in the building that satisfy certain identity attributes. For this purpose, she makes use of a Mobile Node that sends the received RSSIs to the Gateway deployed in the building, so that Alice's position can be calculated. Then, this information is shared through the Context Manager by making use of the bubble concept.

At this point, it should be pointed out that we have considered two alternatives of the main scenario in which the Bubbles Handler can delegate the CP-ABE encryption operation to the platform. To do this, we have considered the addition of a new component, named *CP-ABE Assistant*, which is responsible for encrypting the information before sending to the Context Manager. As already demonstrated in [30, 29], CP-ABE requires expensive cryptographic operations related to the use of pairings. Therefore, in certain situations, the delegation of these operations can alleviate the burden of the end nodes, especially in cases where a huge amount of data needs to be protected. Additionally, we have considered that Alice has associated her Mobile Node (named "Sociotal:UMU:mobilenode:aliceMobilenode") to the bubble "AliceWorkBubble" in the Context Manager, by following the process described in Section 3.5.2.

With these premises, the description of the sequence diagram is as follows. As already described in Section 3.5.1, during the *Phase 1*, users try to obtain the required CP-ABE cryptographic material from the CP-ABE Key Manager. This stage could be carried out at any time before the following interactions. In this case, we have considered Alice and Bob act as a data *Producer* and *Consumer*, respectively. Furthermore, it is also considered that Alice has defined the bubble "AliceWorkBubble" associating her device "SocIoTal:UMU:mobilenode:AliceMobilenode" to it, with a specific policy by following the steps of *Phase 2*. Then, the integration with the localization system is reflected in *Phase 3*. In this case, the Mobile Node worn by Alice sends the set of RSSIs, which are obtained from the reception of beacon messages sent by the anchors. The set of RSSIs is sent to the Gateway (step 3.1) through the WSN topology. After computing the estimated location (step 3.2), the Gateway sends this information to Alice's Bubble Handler (step 3.3). As already mentioned, at this point two alternatives are considered. In the first case (Option 1), Alice's Bubbles Handler runs the CP-ABE encryption algorithm to encrypt Alices's location by using the AliceWorkBubble's policy, and the CP-ABE parameters that are obtained during the Phase 1 (step 3.4). In the second case (Option 2), the CP-ABE encryption is delegated to the CP-ABE Assistant service that is provided by the platform. Towards this end, the Bubble Handler component sends the encryption parameters to such service (3.4.a), so that the encryption operation is performed in the same way (3.4.b). Once the encryption has been completed, the result is sent to

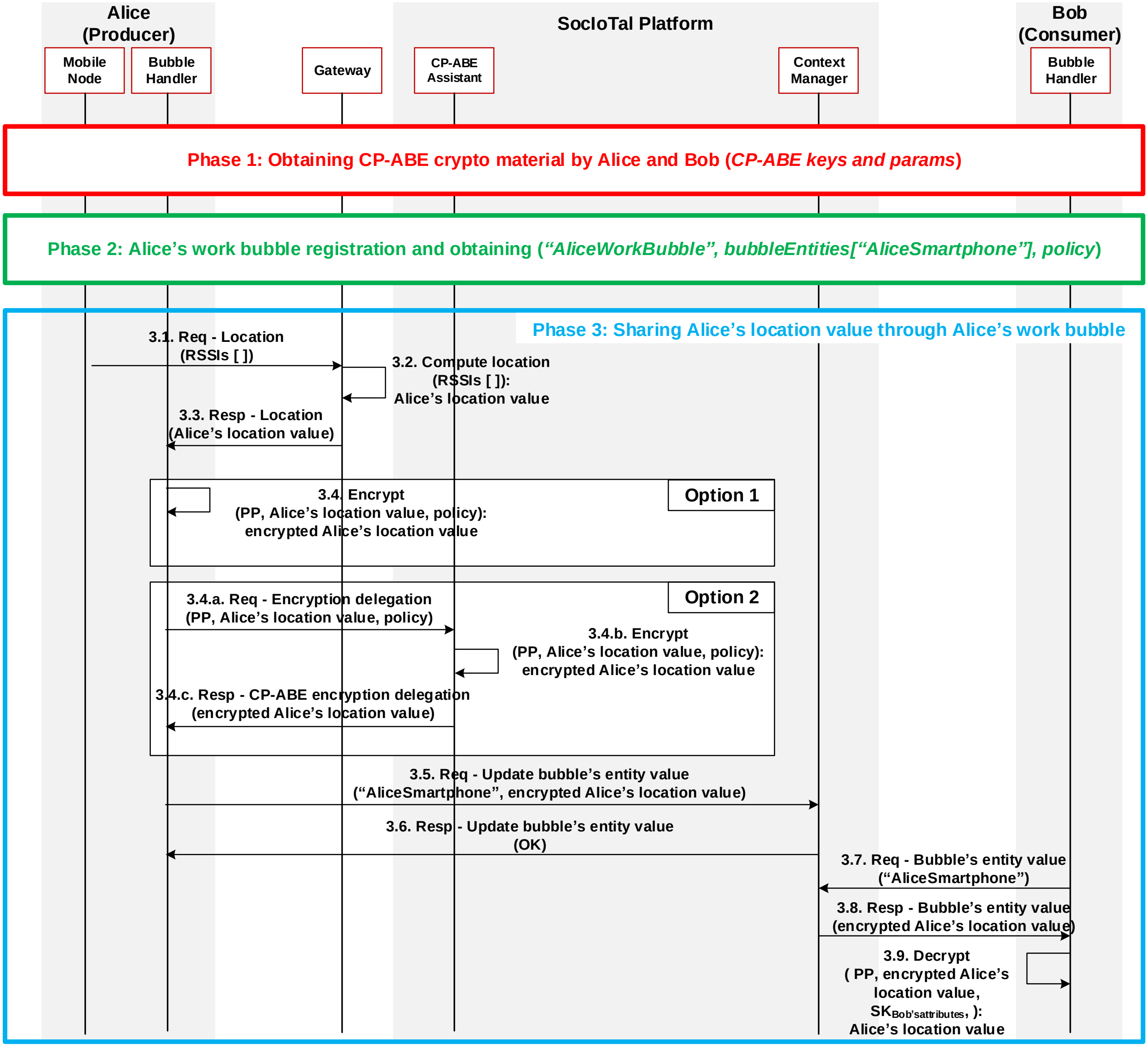

Alice (Producer)
Mobile Node
Bubble Handler
Gateway
SocIoTal Platform
CP-ABE Assistant
Context Manager
Bob (Consumer)
Bubble Handler
Phase 1: Obtaining CP-ABE crypto material by Alice and Bob (CP-ABE keys and params)
Phase 2: Alice's work bubble registration and obtaining ("AliceWorkBubble", bubbleEntities["AliceSmartphone"], policy)
Phase 3: Sharing Alice's location value through Alice's work bubble
3.1. Req - Location (RSSIs [ ])
3.2. Compute location (RSSIs [ ]): Alice's location value
3.3. Resp - Location (Alice's location value)
Option 1
3.4. Encrypt (PP, Alice's location value, policy): encrypted Alice's location value
Option 2
3.4.a. Req - Encryption delegation (PP, Alice's location value, policy)
3.4.b. Encrypt (PP, Alice's location value, policy): encrypted Alice's location value
3.4.c. Resp - CP-ABE encryption delegation (encrypted Alice's location value)
3.5. Req - Update bubble's entity value ("AliceSmartphone", encrypted Alice's location value)
3.6. Resp - Update bubble's entity value (OK)
3.7. Req - Bubble's entity value ("AliceSmartphone")
3.8. Resp - Bubble's entity value (encrypted Alice's location value)
3.9. Decrypt (PP, encrypted Alice's location value, SK$_{Bob'sattributes}$,): Alice's location value

Figure 3.9: Main interactions for sharing location data through bubbles by using the proposed localization system

the requesting Bubble Handler (step 3.4.c).

Then, Alice's Bubbles Handler sends a NGSI *updateContext* message to the Context Manager, in order to update Alice's location of the entity "SocIoTal:UMU:mobilenode:AliceMobilenode" with the encrypted value previously obtained (step 3.5). This message is answered with a confirmation message from the Context Manager (step 3.6). Since then, each query to this entity will return the encrypted value of Alice's location. To do this, Bob can use the NGSI *queryContext* message to get the information associated to "AliceWorkBubble". It should be noted that these steps are optional, since Bob's Bubbles Handler could have previously obtained the information associated to this bubble. Then, this component queries the values associated to "SocIoTal:UMU:mobilenode:AliceMobilenode" through a NGSI *queryContext* method and it obtains the encrypted value of Alice's location (steps 3.7-3.8). In order to decrypt such value, the Bubble Handler runs the CP-ABE decryption algorithm by using Bob's CP-ABE key and the public parameters (step 3.9) that are previously obtained. Additionally, once Alice's location is decrypted, Bob can use the Web-App client (shown in Figure 3.8) to show the Alice's location in the map of the building.

3.7 Use case implementation

Taking into account the previously identified components for the use case, this section is intended to describe the implementation details of each of them. It should be noted that for the implementation of the proposed scenario, we have used the existing implementation of the WUE and Context Manager components. A more detailed description of each component can be found at GitHub[10].

3.7.1 SocIoTal Platform

As described in Section 3.4, the platform aims to embrace different components to enable the creation and management of bubbles in the proposed scenario. By taking into account the addition of the CP-ABE Assistant component for our use case, the implementation details for each component are provided.

3.7.1.0.1 CP-ABE Key Manager It is a HTTPS server implemented in Java servlets accepting requests for obtaining the corresponding CPABE keys and cryptographic parameters. As already mentioned, these keys are associated with the attributes stored in the IdM Service. Towards this end, the implementation integrates a simple Keyrock client library to recover such attributes from that service. For CP-ABE functionality, two different libraries have been employed in order to compare their performance. On the one hand, we have made use of the library provided by [53], which is a Java implementation built on top of the Java Pairing Based Cryptography library (jPBC) [54]. On the other hand, we have deployed the required functionality for CP-ABE aspects through the use of the library provided in [55], which is a C implementation (based on the PBC library [56]) that was developed by the authors of the CP-ABE scheme.

3.7.1.0.2 IdM Service This service has been implemented as a modified deployment of the FIWARE Keyrock IdM system. In particular, the KeyRock server has been extended to consider attributes of the SCIM standard that was not included in the original version. It should be noted that a more complete version of this service including privacy-preserving features has been extended in the scope of the SocIoTal project, as described in [8].

3.7.1.0.3 CP-ABE assistant It is a HTTPS server implemented in Java servlets accepting requests for encrypting data by using CP-ABE. Like in the case of the CP-ABE Key Manager, the already mentioned CP-ABE libraries have been employed in order to compare their performance in the encryption process.

3.7.2 Bubbles Handler

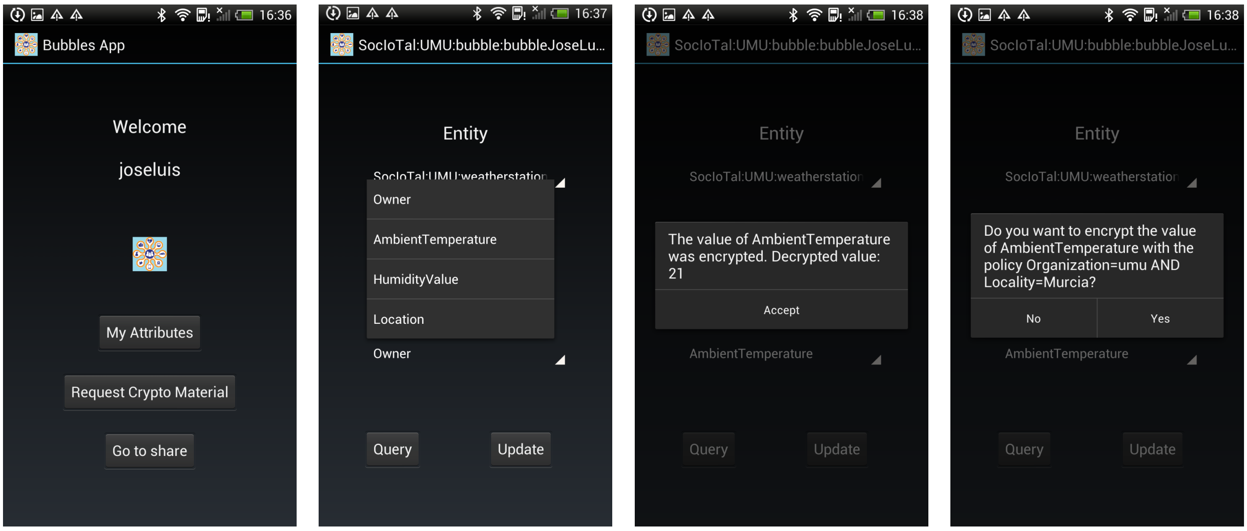
This represents the main component to manage the secure data sharing aspects through the bubbles, as well as to obtain the required CP-ABE cryptographic material. It integrates a Keyrock Client library that provides a basic API for identity management by implementing a client to interact with the IdM Service for authentication purposes. It also integrates an OMA-NGSI client library in order to ease the management of the messages that are sent to the Context Manager, based on the use of the *gson* library[11]. Like in the case of the previous components, we have tested two different libraries to perform the required CP-ABE encryption/decryption operations ([53], [55]).

It should be noted that, while these libraries can be executed in any Java-compatible device, Figure 3.10 shows some examples of screenshots related to the deployment on Android devices. Additional information about this component can be found at Github [12]

[10] http://github.com/sociotal/SOCIOTAL/wiki

[11] https://github.com/google/gson

[12] https://github.com/sociotal/SOCIOTAL/wiki/SocIoTal-Group-Manager



The value of AmbientTemperature was encrypted. Decrypted value: 21

Do you want to encrypt the value of AmbientTemperature with the policy Organization=umu AND Locality=Murcia?

Figure 3.10: Bubbles Handler screenshot examples

3.7.3 Gateway

3.7.3.0.1 Implementation of the radio-location metrics on the gateway The implementation of the location-dependent attribute algorithm related to Section 3.6.1 in Java is detailed on Listing 3.9. It could be implemented in any device that supports Java. For SocIoTal trials, it runs in the local Gateway that first estimates the distance of each anchor from the sink node from the collected RSSI values.

Listing 3.9: Code for the computation of the distances

```
private double distance(double rssi){
        //d = (d_ref x 10^(rssi_ref − rssi) / (10 x a))^2
        return Math.pow(REF_DISTANCE * Math.pow(10, (REF_RSSI − rssi) / (10 * ALPHA)), 2);
}
```

3.7.3.0.2 Practical implementation of the positioning algorithm Moreover, as practical implementation of the algorithm described in Section 3.6.2, the fixed anchors are calibrated and their relative coordinates in the given environment is estimated as depicted by the code detailed on Listing 3.10. Given this calibration, Listing 3.11 depicts the Java implementation of the Gradient-Descent technique used to estimate the relative 2D coordinates of a sink node in the real environment.

Listing 3.10: Code for the anchor calibration

```
static{
        //Fill in the anchor map
        anchorMap.put("00124b00060d85b3", new Coord(0.00, 0.00));
        anchorMap.put("00124b00060d84c4", new Coord(26.00, 0.00));
        anchorMap.put("00124b00060d8475", new Coord(46.00, 0.00));
        anchorMap.put("00124b00060d8473", new Coord(54.00, 65.00));
        anchorMap.put("00124b00060d8480", new Coord(25.00, 65.00));
        anchorMap.put("00124b00060d84ba", new Coord(0.00, 65.00));
        anchorMap.put("00124b00060d84f5", new Coord(0.00, 38.00));
        anchorMap.put("00124b00060d84aa", new Coord(23.00, 34.00));
}
```

Listing 3.11: Code for the computation of the coordinates

```
public Coord computeLeastMeanSquare() {
        //Mapping of anchor address <=> mean RSSI with this anchor
        Map< String , Double> meanRssi.size();
        if (size < 3)
          //We cannot compute localization with less
          //than three anchors
          return null;

        final List<String> macs =
          new ArrayList<>(meanRssi.keySet());
        //Choose the first anchor as an arbitrary reference
        //for computations
        final String firstMac = macs.get(0);
        final Coord firstCoord = anchorMap.get(firstMac);
        final double firstDistance =
          distance(meanRssi.get(firstMac));

        //Fill the matrices
        Matrix a = new Basic1DMatrix(size, 2);
        Matrix h = new Basic1DMatrix(size, 1);
        for(int i = 1; i < meanRssi.size(); i++){
          final String mac = macs.get(i);
          final Coord coord = anchorMap.get(mac);

          //       |x_i - x_0       y_i - y_0 |
          // A = |                            |
          //       |x_k - x_0       y_k - y_0 |

          a.set(i, 0, coord.x - firstCoord.x);
          a.set(i, 1, coord.y - firstCoord.y);

          //       |x_i^2 - x_0^2 + y_i^2 - y_0^2 + d_0^2 - d_i^2|
          // h = |                                                |
          //       |x_k^2 - x_0^2 + y_k^2 - y_0^2 + d_0^2 - d_k^2|
          //        |--------------------------|
          //                      hprime

          final double hprime = Math.pow(coord.x, 2) -
            Math.pow(firstCoord.x, 2) +
            Math.pow(coord.y, 2) -
            Math.pow(firstCoord.y, 2);
          h.set(i, 0, hprime, firstDistance -
            distance(meanRssi.get(mac)));
        }
}
```

Once estimated on the local gateway thanks to this implementation of the gradient-based method, the time-stamped 2D $(\tilde{x}_m(t), \tilde{y}_m(t))$ coordinates produced at time t can be sent to the Bubbles Handler component for encryption purposes before sending to the Context Manager.

3.7.4 Web App environment

As part of the deployment of the radio localization system (see Figure 3.8), a localization Web App has been developed and it is shown in Figure 3.11. The Web App provides a map of the building where the active anchors - those whom the RSSI is taken into account by the sink node - are colored in blue. The inactive anchors are colored in pastel blue and the sink node (in this case, worn by Alex) makes reference to a person moving on the building. His displacement is followed on the map thanks to the web app. This way, in case of a successful CP-ABE decryption process, a user can employ this app to visualize the position of other users in the building.

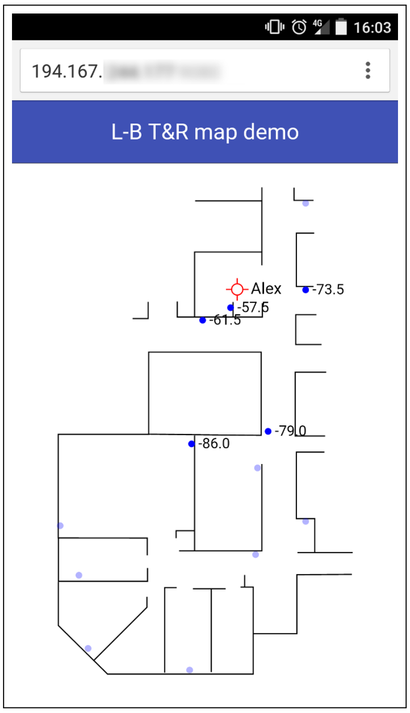


Figure 3.11: Web App screenshot

3.8 Evaluation and Performance Results

After describing the implementation details of the different components, this section provides a description of the specific hardware, as well as the results obtained from the different processes previously explained.

3.8.1 Hardware features

For the radio localization process, the OpenMote-CC2538 hardware technology is used for the low-level devices. The OpenMote-CC2538 is a System-on-Chip embedding a Cortex-M3 microcontroller and a CC2520-like radio transceiver from Texas Instrument compliant for 2.4GHz IEEE 802.15.4-2006 standard. The microcontroller runs up to 32 MHz and includes 32 Kbytes of RAM and 512 Kbytes of Flash, as well as the usual peripherals. It embeds a battery monitor, temperature and lighting sensors, as well as a 3-axes accelerometer. This sensor mote is designed for IoT applications. For this experiment, the latest version of Contiki 3.x is embedded as Operating System. In the following the metrics used to estimate the coordinate of a sink node from the collected RSSI measure from the fixed anchors are detailed both with the analytic formula and the corresponding implementation in the local gateway. The experiments have been performed with the OpenMote hardware that are narrow-band IEEE 802.15.4 devices operating at 2.4Ghz in the ISM band. Furthermore, the gateway functionality was implemented on an Intel(R) Xeon(R) CPU E3-1220 v3 @ 3.10GHz.

Moreover, the Bubbles Handler has been implemented on an Android device Samsung Galaxy J3 (6) with a 1.5 GHz quad-core processor, 1.5 GB of RAM and version 5.1 of Android (Lollipop). In addition, the platform components (besides the Context Manager and the WUE) were deployed on an Intel Core i5-5257U 2.70 GHz processor and 2GB of RAM with Ubuntu 14.

3.8.2 Considerations about bubbles scenario evaluation

As previously described, we have made use of two different implementations for the CP-ABE functionality. Since the Bubbles Handler is deployed on an Android device, it should be noted that, for the C library, we have followed a similar approach to [30] by using *Android Native Development Kit* (NDK) and CMake[13] tools, in order to accommodate C code into the Android application. By using both implementations, we have made use of the following aspects:

- **Hardware**. As described in Section 3.6, the CP-ABE encryption operation could be executed on the Bubbles Handler or the CP-ABE Assistant within the platform. Consequently, evaluation results consider different hardware devices for the CP-ABE encryption process.
- **Performance parameters**. In addition to the delay required for the main CP-ABE algorithms, energy consumption is also considered for evaluation purposes in the case of the Bubbles Handler, which is deployed on Android devices. For this purpose, we have made use of the tool Powertutor [57], which is also used in previous works [30].
- **Number of bubbles' attributes**. As stated at Section 3.4, we use up to 6 different SCIM attributes to be employed for restricting the access to data within a specific bubble. While additional attributes could be used, it would make attribute management more difficult to be performed.
- **Security level**. Both implementations use type A pairings based on the curve $y^2 = x^3 + x$ over the field F_p for some prime $p = 3 \ mod \ 4$. The security strength of the scheme can be tuned by modifying two parameters: the size of the field p, and the prime order q [30, 29]. In our case, evaluation is performed by considering 80-bit (i.e. $|p| = 512$, $|q| = 160$) and 112-bit (i.e. $|p| = 1024$, $|q| = 224$) security levels.
- **Scalability**. While previous aspects are already considered by [30] and [58], we assess the impact of the concurrent execution of CP-ABE cryptographic algorithms. Beyond the use of location data, we consider this aspect crucial due to the potential huge amount of data to be protected on an IoT-enabled scenario.

Additionally, it should be noted that for CP-ABE encryption and decryption evaluation, only CP-ABE policies with AND gates and 1-level policy tree are considered. This is due to the implicit bubble definition in which all the attributes in the bubble are used to restrict the access to data within a specific bubble. While a more expressive (and consequently, more complex) definition could be used, the proposed bubble specification aims to provide a simple but still consistent view so that end users are enabled to control the disclosure of their devices' data. In addition, each experimentation result was obtained as the mean value of 10 executions, and confidence intervals were calculated by considering a 95% confidence level.

3.8.3 Obtaining CP-ABE cryptographic material

Taking into account these premises, Figure 3.12 shows the delay required for both implementations in the case of CP-ABE key generation that is performed on the CP-ABE Key Manager. Indeed, this process makes reference to the step 1.5 in Figure 3.3). The x-axis represent the number of attributes that is associated to the generated key. Firstly, for a 1-operation workload, the implementation based on [53] requires 108 ms to generate a 1-attribute key (379 ms for a 6-attribute key) in case of 80-bits security level. This delay grows until 449 ms and 2176 ms when considering a 112-bits security level. This time is considerably reduced by using the implementation based on [55] in which the delay obtained is 78 ms and 368 ms for this security level.

Secondly, for a 25-operation workload, the difference is more considerable; the Java library requires 1253 ms to generate 1-attribute key (7097 ms for a 6-attribute key) in case of 80-bits security level (5973 ms/33484 ms for 112-bits). For the C library, this delay is reduced to 2167ms/11221 ms for 112-bits security level, which represents a 63,72% and 66,49% decrease respectively. While it represents a considerable better performance, it should be noted that this process is not expected to be done frequently, since CP-ABE keys are associated to users' attributes, which does not affect possible dynamic changes with CP-ABE policies or the creation of new bubbles in our scenario.

[13] https://cmake.org/

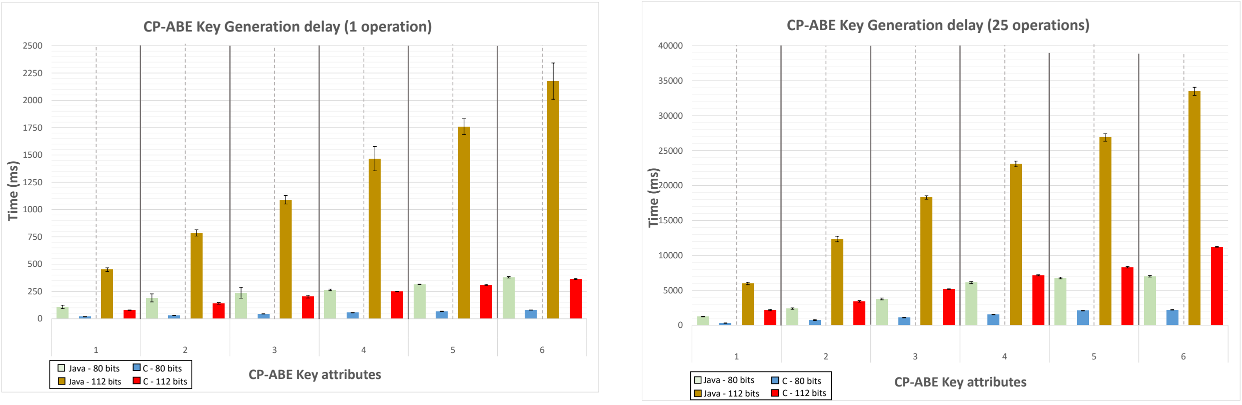


Figure 3.12: CP-ABE Key Manager performance

Location computations (Values in μs)					
Barycenter method	1000000 samples	min 1.0	max 5328.0	mean 1.18	stddev 7.4
Least mean square method	1000000 samples	min 7.0	max 54536.0	mean 7.8	stddev 57.2
Gradient descent method	10000 samples	min 2432.0	max 89195.0	mean 2569.3	stddev 1042.2

Table 3.1: Performance of the location algorithms in the Gateway

3.8.4 Detecting and sending location data to gateway

Regarding the interactions related to the localization system, once the data collected by the Mobile Node, aggregated messages including the RSSI measures are sent to the Gateway (step 3.1 in Figure 3.9). This operation takes few milliseconds (between 1ms and 100ms) in a scope that trigs on the mote transmission and includes the receiving and the time of raising the IEEE 802.15.4 stack in the Linux environment on the gateway.

3.8.5 Data processing on the gateway

The data processing (step 3.2 in Figure 3.9) consists of performing the distances for each collected RSSI using the barycenter method. This processing takes few μs as depicted in the table below. For these measurements, the statistics are obtained from a collection of 1 million of samples uniformly distributed on the considered interval. Once the distances evaluated, the gradient descent method is used to estimate the 2D coordinates of the sink mote. This technique takes significantly more time than the least mean square method, as depicted in the Table 3.1. Nevertheless, the precision, the consistency and the reproducibility of this technique is higher and justifies its use to provide to the end user of relevant estimation of the sink localization.

3.8.6 Encrypting location data through bubbles

Taking into account the proposed alternatives for CP-ABE encryption (and delegated encryption), in this section we compare both implementations when they are executed on different hardware.

On the one hand, Figure 3.13 shows the delay required for the encryption by fluctuating the workload (from 1 to 5 operations). In particular, these results refer to the step 3.4 in Figure 3.9. In case of a 1-operation workload, the implementation based on the Java library requires 1232 ms/5283 ms for 1 and 6-attribute policies when a 80-bits security level is considered. This time is around 3799 ms/20006 ms in the case of a 112-bits security level, representing a 208,36%/278,69% increase. For this operation and the same security level, the C library requires 351 ms/1355 ms, which means a 90,76%/93,23% decrease regarding the previous library. Furthermore, in case of a 5-operation workload, the implementation based on C represents a 92,37%/95,33% decrease by considering the same configuration.

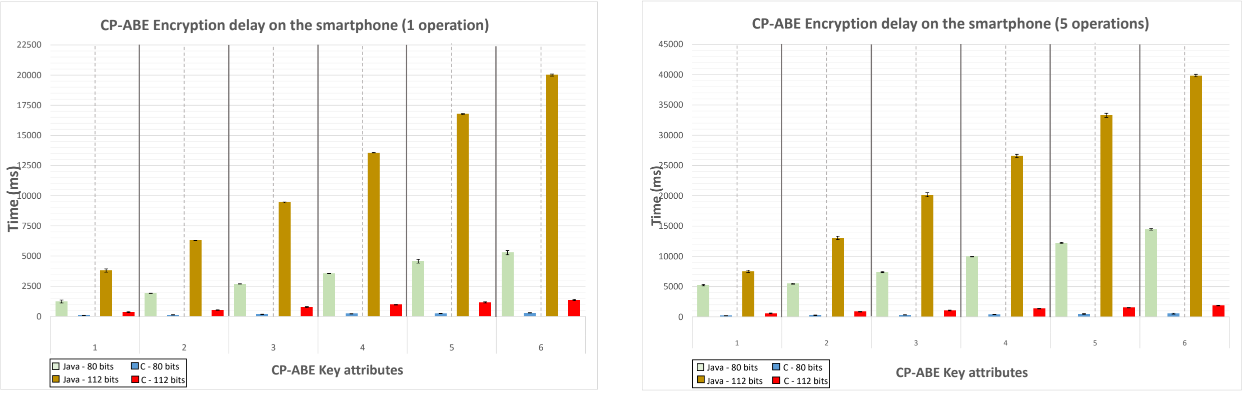


Figure 3.13: CP-ABE Encryption delay on the Bubbles Handler

On the other hand, Figure 3.14 depicts the performance of the CP-ABE encryption operation when it is executed by the CP-ABE Assistant (3.4.b). It should be noted that, in this case, we have considered a workload up to 50 operations, since this component will be in charge of encrypting data coming from different devices in the building. In this case, for a 1-operation workload, the Java implementation needs 468 ms/2370 ms for 1 and 6-attribute policies when a 112-bits security level is considered. In this sense, the C library requires 74 ms/363 ms, so a 84,19%/84,68% decrease regarding the previous library. It should be noted that, since CP-ABE encryption is performed more efficiently in the platform, this decrease is less significant than in the case of the smartphone. In case of a 50-operation workload, the Java library requires 16126 ms/61639 ms for 1 and 6-attribute policies and a 112-bits security level. For the C library, this delay is reduced to 4125ms/19741 ms, so a 74,42% and 67,97% decrease is obtained respectively. In general, while evaluation results are obviously better in the case of the CP-ABE Assistant, it should be noted that this entity would be able to access the data to be encrypted. Consequently, depending on the security and performance requirements of a specific scenario, the inclusion of this component could be appropriate or not.

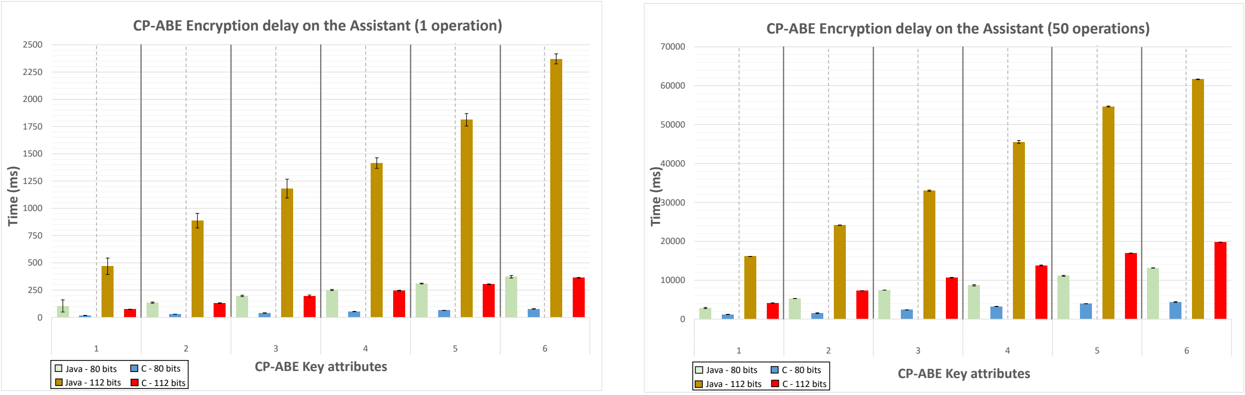


Figure 3.14: CP-ABE Encryption delay on the CP-ABE Assistant

Moreover, the aspects related to energy consumption are a major concern for scenarios where certain devices (e.g. a smartphone) are employed. Because of it, we provide a set of experimental results regarding the energy consumption for the CP-ABE encryption operation in case of the Bubbles Handler is running this operation. These results are shown in Figure 3.15 by considering a workload between 1 and 5 operations. According to it, the encryption operation requires between 3,2 J and 14,8 J in the case of the Java implementation for 1 and 6-attributes policies, and a 112-bits security level. In case of the C implementation, this consumption is reduced to 0,9 J and 2 J meaning a 71,88% and 86,49% decrease. In addition, for a 5-operations workload, the Java library consumes 15,1 J / 65,3 J, while the C implementation requires 1,7 J / 4,1 J, which represents an

88,74%/93,72% decrease.

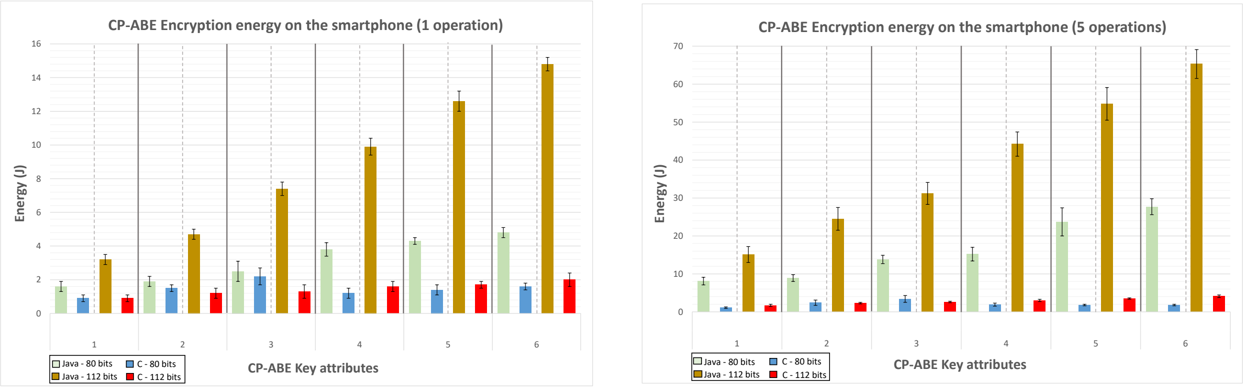


Figure 3.15: CP-ABE Encryption energy consumption on the Bubbles Handler

3.8.7 Decrypting location data through bubbles

When encrypted data are obtained through the Context Manager, they need to be decrypted by using the CP-ABE decryption algorithm (step 3.9 in Figure 3.9). For this operation, Figure 3.16 shows the time required by the Bubbles Handler by considering a workload of 1 and 5 operations. In this case, unlike the encryption, we have considered the decryption process should not be delegated to any other entity, since the CP-ABE key would be disclosed, so that such entity would have access all the information being shared.

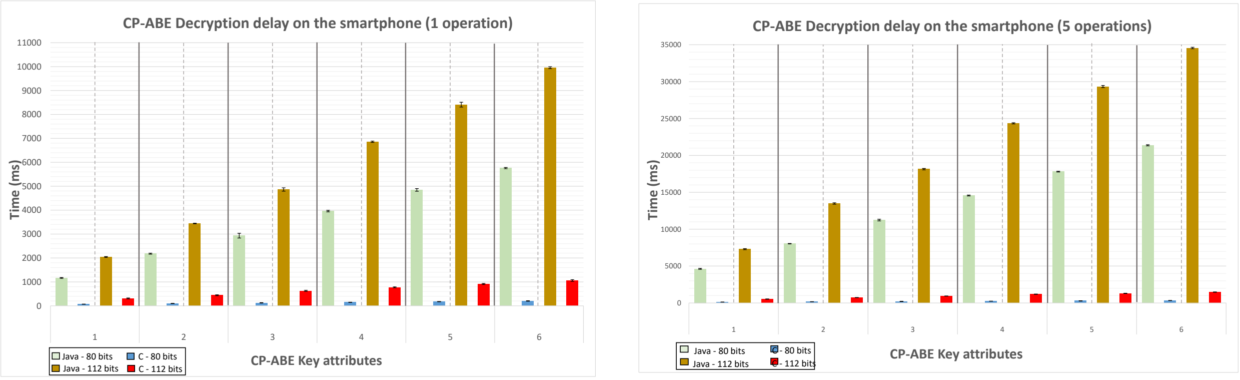


Figure 3.16: CP-ABE Decryption delay on the Bubbles Handler

According to results, in case of a 1-operation workload, the Java-based implementation library requires 1165 ms/5758 ms for 1 and 6-attribute policies when a 80-bits security level is considered. This delay is increased to 2041 ms/9956 ms in the case of a 112-bits security level, representing a 75,19%/72,91% increase. For this operation and the same security level, the C library requires 309 ms/1059 ms, which means a 73,48%/89,36% decrease regarding the previous library. Furthermore, in case of a 5-operation workload, the implementation based on C obtains 526 ms/1476 ms (7297 ms/34559 ms in case of the Java library), which represents a 92,79%/95,73% respectively.

For energy consumption, like in the case of encryption, values remain almost constant in case of the implementation based on the C library. Indeed, it requires 0,3 J/1 J for 1 and 6-attribute policies when a 80-bits security level is considered. This consumption is around 0,9 J/1,8 J in the case of a 112-bits security level. In case of the Java implementation, the energy consumption grows up to 1,3-6 J for a 80-bits security level and 1,9-13,6 J by considering a 112-bits level. Finally, in case of a 5-operation workload and this security level, the Java library

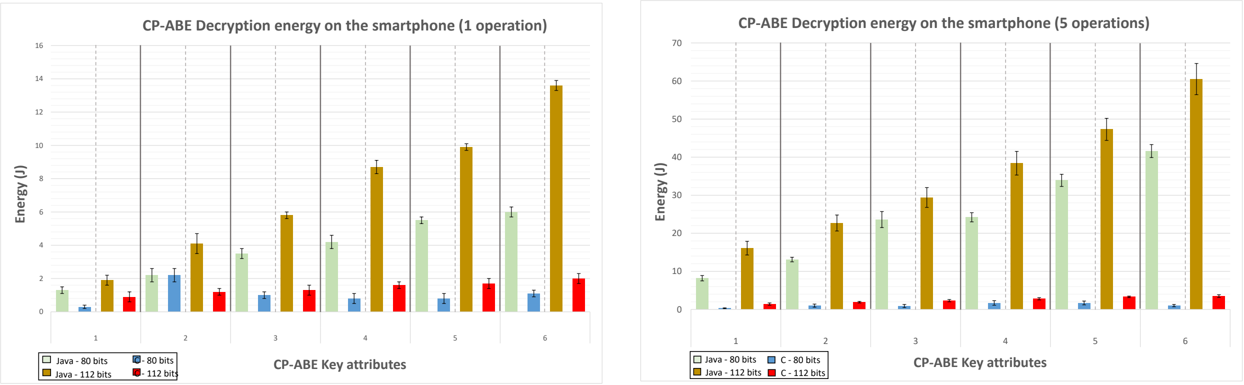


Figure 3.17: CP-ABE Decryption energy consumption on the Bubbles Handler

consumes 16,1-60,5 J, while the implementation based on C requires 1,4-3,5 J, which represents a 1050% a 91,3%/94,21% decrease by considering the same configuration.

Such results demonstrate the proposed approach could be used even when a huge amount of personal data needs to be protected through the use of bubbles. Indeed, as the number of concurrent operations grows, the performance improvement by using the library from [55] is still more significant, providing practical results even when considering 6-attributes policies and a 112-security level. With the proposed approach, while the integration of CP-ABE on smartphone devices requires heavy cryptographic operations to be performed on end devices, confidentiality is ensured on an end-to-end basis. However, it should be noted that the Bubbles Handler component could be instantiated by a different entity (beyond the use of a smartphone). In this case, the delegation of the CP-ABE encryption into the assistant component could be suitable at the expense of reducing the security restrictions in the scenario. In this sense, the application of such delegation approach could be analyzed and compared by considering more constrained IoT devices, such as in [58].

3.9 Summary

Current trends in smart cities promote the development of new services based on sharing massive amounts of information, representing the cornerstone of a data-driven economy. In these scenarios, the specification and enforcement of security preferences on shared data is an extremely challenging aspect, which must be addressed by comprehensive approaches considering aspects such as flexibility, scalability, heterogeneity and usability. This work has presented the concept of bubble as a way to protect devices' data by considering potential relationships between their owners. Toward this end, the use of the CP-ABE scheme provides significant benefits in terms of key management to provide confidentiality even when this information needs to be shared through central data platforms. The resulting approach has been further deployed on different hardware components, as well as the FI-WARE platform in order to demonstrate their applicability on a real environment. In order to complement our proposal, future work is mainly focused on the integration of anonymous signature schemes to protect the integrity of shared data within a bubble in a privacy-preserving way.

4 Conclusions

This deliverable takes as starting point the requirements pointed in D1.3, as well as the C-ITS and Smart objects frameworks proposed in D3.2. The aim at the end, it is to provide new security proposals based on such frameworks to be deployed on real IoT scenarios. Specifically, we have introduced two different solutions. On the one hand, we have propose two schemes, which are tailored to the needs of C-ITSs, in order to improve privacy aspects and provide anonymous authentication. Both schemes may be employed in combination or independently as well. On the other hand, we have also presented a Smart objects framework-based solution intended to protect data by considering certain preferences of their owners. To this end, this proposal integrates the CP-ABE scheme, along with certain functionality provided by FI-WARE enablers, to achieve secure communications between groups of entities through central data platforms.

Bibliography

[1] C. Freitag, J. Katz, and N. Klein, “Symmetric-key broadcast encryption: The multi-sender case,” in *International Conference on Cyber Security Cryptography and Machine Learning*. Springer, 2017, pp. 200–214.

[2] B. Chor, A. Fiat, and M. Naor, “Tracing traitors,” in *Annual International Cryptology Conference*. Springer, 1994, pp. 257–270.

[3] D. Boneh and M. Naor, “Traitor tracing with constant size ciphertext,” in *Proceedings of the 15th ACM conference on Computer and communications security*. ACM, 2008, pp. 501–510.

[4] J. Petit, F. Schaub, M. Feiri, and F. Kargl, “Pseudonym schemes in vehicular networks: A survey,” *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 228–255, 2015.

[5] A. Caragliu, C. D. Bo, and P. Nijkamp, “Smart cities in Europe,” *Journal of Urban Technology*, vol. 18, no. 2, pp. 65–82, apr 2011.

[6] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2005, pp. 457–473.

[7] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute-Based Encryption,” in *2007 IEEE Symposium on Security and Privacy*. IEEE, may 2007.

[8] J. B. Bernabe, J. L. Hernandez-Ramos, and A. F. S. Gomez, “Holistic Privacy-Preserving Identity Management System for the Internet of Things,” *Mobile Information Systems*, vol. 2017, pp. 1–20, 2017.

[9] M. Bauer, E. Kovacs, A. Schulke, N. Ito, C. Criminisi, L.-W. 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*. IEEE, oct 2010.

[10] A. Martinez-Balleste, P. A. Pérez-Martínez, and A. Solanas, “The pursuit of citizens’ privacy: a privacy-aware smart city is possible,” *Communications Magazine, IEEE*, vol. 51, no. 6, pp. 136–141, 2013.

[11] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, “Evaluating critical security issues of the IoT world: Present and Future challenges,” *IEEE Internet of Things Journal*, pp. 1–1, 2017.

[12] A. Sedrati and A. Mezrioui, “Internet of Things challenges: A focus on security aspects,” in *2017 8th International Conference on Information and Communication Systems (ICICS)*. IEEE, apr 2017.

[13] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, “Data Mining for Internet of Things: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 77–97, 2014.

[14] J.-M. Bohli, R. Kurpatov, and M. Schmidt, “Selective decryption of outsourced IoT data,” in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, dec 2015.

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM conference on Computer and communications security - CCS*. ACM Press, 2006.

[16] R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-based encryption with non-monotonic access structures,” in *Proceedings of the 14th ACM conference on Computer and communications security - CCS*. ACM Press, 2007.

[17] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. de Panafieu, and C. Ràfols, “Attribute-based encryption schemes with constant-size ciphertexts,” *Theoretical Computer Science*, vol. 422, pp. 15–38, mar 2012.

[18] G. Wang, Q. Liu, and J. Wu, “Hierarchical attribute-based encryption for fine-grained access control in

cloud storage services," in *Proceedings of the 17th ACM conference on Computer and communications security - CCS*. ACM Press, 2010.

[19] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, jan 2013.

[20] Z. Wan, J. Liu, and R. H. Deng, "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, apr 2012.

[21] L. Touati, Y. Challal, and A. Bouabdallah, "C-CP-ABE: Cooperative Ciphertext Policy Attribute-Based Encryption for the Internet of Things," in *2014 International Conference on Advanced Networking Distributed Systems and Applications*. IEEE, jun 2014.

[22] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," *Future Generation Computer Systems*, vol. 49, pp. 104–112, aug 2015.

[23] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[24] G. Bianchi, A. T. Capossele, C. Petrioli, and D. Spenza, "AGREE: exploiting energy harvesting to support data-centric access control in WSNs," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2625–2636, nov 2013.

[25] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," in *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE, apr 2015.

[26] D. Thatmann, S. Zickau, A. Förster, and A. Küpper, "Applying attribute-based encryption on publish subscribe messaging patterns for the internet of things," in *Data Science and Data Intensive Systems (DSDIS), 2015 IEEE International Conference on*. IEEE, 2015, pp. 556–563.

[27] "Information technology. message queuing telemetry transport (MQTT) v3.1.1."

[28] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 763–771, may 2014.

[29] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of attribute-based encryption: Toward data privacy in the iot," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 725–730.

[30] M. Ambrosin, M. Conti, and T. Dargahi, "On the feasibility of attribute-based encryption on smartphone devices," in *Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*. ACM, 2015, pp. 49–54.

[31] K. Grizzle, E. Wahlstroem, and C. Mortimore, "System for cross-domain identity management: Core schema," Tech. Rep., sep 2015.

[32] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. L. Gall, and M. Zhao, "Standards-Based Worldwide Semantic Interoperability for IoT," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 40–46, dec 2016.

[33] I. Elicegui, C. López, L. Sánchez, J. Lanza, L. Muñoz, A. Pintus, A. Manchinu, and A. Serra, "Design and Implementation of a Cloud-Based Platform for Unleashing the Personal and Communal Internet of Things," *Mobile Information Systems*, vol. 2017, pp. 1–14, 2017.

[34] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The Social Internet of Things (SIoT)–When social networks meet the Internet of Things: Concept, architecture and network characterization," *Computer Networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[35] O. Garcia-Morchon, S. L. Keoh, S. Kumar, P. Moreno-Sanchez, F. Vidal-Meca, and J. H. Ziegeldorf, "Securing the IP-based Internet of Things with HIP and DTLS," in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks - WiSec*. ACM Press, 2013.

[36] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology — CRYPTO 2001*. Springer Berlin Heidelberg, 2001, pp. 213–229.

[37] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen, and T. Scavo, "Security Assertion Markup Language (SAML) V2.0 Technical Overview," 2008.

[38] D. Hardt, "RFC 6749: The Oauth 2.0 Authorizaion Framework," *Internet Engineering Task Force (IETF)*, 2012.

[39] J. B. Bernabe, I. Elicegui, E. Gandrille, N. Gligoric, A. Gluhak, C. Hennebert, J. L. Hernandez-Ramos, C. Lopez, A. Manchinu, K. Moessner, M. Nati, C. O'Reilly, N. Palaghias, A. Pintus, L. Sanchez, A. Serra, and R. van Kranenburg, "SocIoTal — the development and architecture of a social IoT framework," in *2017 Global Internet of Things Summit (GIoTS)*. IEEE, jun 2017.

[40] "Open mobile alliance (oma) ngsi context management v1.0," 2010. [Online]. Available: http://www.openmobilealliance.org/release/NGSI/V1_0-20120529-A/OMA-TS-NGSI_Context_Management-V1_0-20120529-A.pdf

[41] J. L. Hernandez-Ramos, J. B. Bernabe, and A. Skarmeta, "ARMY: architecture for a secure and privacy-aware lifecycle of smart objects in the Internet of My Things," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 28–35, 2016.

[42] J. Camenisch and E. V. Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proceedings of the 9th ACM conference on Computer and communications security - CCS*. ACM Press, 2002. [Online]. Available: https://doi.org/10.1145%2F586111.586114

[43] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security - ASIACCS*. ACM Press, 2010.

[44] S. Perez, J. L. Hernandez-Ramos, S. N. Matheu-Garcia, D. Rotondi, A. F. Skarmeta, L. Straniero, and D. Pedone, "A Lightweight and Flexible Encryption Scheme to Protect Sensitive Data in Smart Building Scenarios," *IEEE Access*, vol. 6, pp. 11 738–11 750, 2018.

[45] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User Collusion Avoidance CP-ABE With Efficient Attribute Revocation for Cloud Storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, jun 2018.

[46] E. P. Paper, "Smart buildings: Energy efficiency first!" http://euroace.org/wp-content/uploads/2015/10/EA_Smart_Buildings_Feb_2017_Final.pdf, 2017, [Online; accessed 20-December-2017].

[47] C. Hennebert and J. D. Santos, "Security Protocols and Privacy Issues into 6LoWPAN Stack: A Synthesis," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 384–398, oct 2014.

[48] P. Coronel, S. Furrer, W. Schott, and B. Weiss, "Indoor Location Tracking Using Inertial Navigation Sensors and Radio Beacons," in *The Internet of Things*. Springer Berlin Heidelberg, pp. 325–340.

[49] M. Laaraiedh, S. Avrillon, and B. Uguen, "Enhancing Positioning Accuracy through Direct Position Estimators Based on Hybrid RSS Data Fusion," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*. IEEE, apr 2009.

[50] J. J. Nielsen, H. Noureddine, N. Amiot, B. Uguen, M. Laaraiedh, I. Raos, I. Arambasic, V. Savic, S. Z. Bello, J. Dominguez, B. Denis, and R. Raulefs, "Assessment of cooperative and heterogeneous indoor localization algorithms with real radio devices," in *2014 IEEE International Conference on Communications Workshops (ICC)*. IEEE, jun 2014.

[51] B. Denis, L. He, and L. Ouvry, "A Flexible Distributed Maximum Log-Likelihood Scheme for UWB Indoor Positioning," in *2007 4th Workshop on Positioning, Navigation and Communication*. IEEE, mar 2007.

[52] G. Destino, D. Macagnano, G. Abreu, B. Denis, and L. Ouvry, "Localization and Tracking for LDR-UWB Systems," in *2007 16th IST Mobile and Wireless Communications Summit*. IEEE, jul 2007.

[53] Y. Wang, "Cp-abe java implementation," 2013. [Online]. Available: https://github.com/junwei-wang/cpabe

[54] A. D. Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *2011 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, jun 2011.

[55] J. Bethencourt, A. a. r. Sahai, and B. a. r. Waters, "cpabe toolkit," 2011. [Online]. Available: http://acsc.cs.utexas.edu/cpabe/

[56] B. Lynn, "Pbc library," 2006. [Online]. Available: http://crypto.stanford.edu/pbc

[57] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis - CODES/ISSS*. ACM Press, 2010.

[58] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R. Moosavi, A. M. Rahmani, and P. Liljeberg, "On the Feasibility of Attribute-Based Encryption on Internet of Things Devices," *IEEE Micro*, vol. 36, no. 6, pp. 25–35, nov 2016.