Opportunistic Smart Object Aggregation based on Clustering and Event Processing

Fernando Terroso-Saenz, José L. Hernández-Ramos, Jorge Bernal Bernabe, Antonio F. Skarmeta
Department of Information and Communications Engineering
Computer Science Faculty
University of Murcia, Spain
{fterroso, jluis.hernandez, jorgebernal, skarmeta}um.es

***Abstract*—In the envisioned Internet of Things ecosystems, Smart objects are intended to create groups of devices in order to provide higher level services to be leveraged by citizens. However, because of the dynamic nature of such scenarios, the discovery, management and operation of such dynamic coalitions taking into account security and privacy concerns, is a challenging task that has not been properly addressed yet. In this sense, the present proposal devises a novel approach to automatically compose opportunistic aggregations of objects (*bubbles*) based on Complex Event Processing (CEP) and fuzzy clustering. While the former detects certain events that could give raise to discover new bubbles, the latter allows compose aggrupations of similar objects acting as candidate bubbles. Finally, the application of the proposal in an educational domain is put forward.**

***Keywords*-IoT, smart object relationship, Big Data, Clustering, CEP**

I. Introduction

The Internet of Things (IoT) [1] paradigm promotes a global network of heterogeneous and autonomous devices, which are intended to interact each other for the realization of innovative and valuable services in Smart Cities ecosystems[2]. Because of the huge scale and inherent nature of entities composing such scenarios, it is envisioned that such devices often operate as a group, in order to perform more complex tasks that cannot be fulfilled by a single device. Furthermore, given the dynamism and pervasiveness of IoT scenarios, interaction among devices or Smart Objects [3] without a predefined trust relationship will be required. In this sense, smart objects should be able to set up autonomously dynamic groups based on the context and their owner's preferences, while security and privacy are properly managed. This drives the need of new self-managing models to allow IoT smart objects to setting up groups and establishing trust relationships among each other, while dealing with inherent security and privacy concerns.

In this sense, the emerging Social Internet of Things paradigm (SIoT) [4] promotes the interaction among smart objects, based on different kinds of relationships that can be established among such devices. SIoT represents a potential approach for the establishment of innovative networking techniques by integrating social network aspects to the smart object world, in order to drive how information is shared among IoT devices.However, the vast amount of data generated and shared by millions of such smart objects makes the creation, management and discovery of potential groups of smart objects a great challenge that needs to be properly addressed. Furthermore, security and privacy implications must be tackled in order the potential of such dynamic coalitions can be exploited in the design of higher level services.

Towards this end, this work proposes different mechanisms in order to deal with the different stages concerning the life cycle of groups of smart objects, such as their creation, discovery or management, as well as their operation. We consider the concept of *bubble* of smart objects as a group of devices that are intended to share information under a common set of security and privacy restrictions. 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. Specifically, this work relies on different techniques, such as Complex Event Processing (CEP) [5] and clustering algorithms for bubble discovery and management. CEP allows real time context processing based on rules, enabling efficient management of large amount of events linked to the smart objects context. It can be applied, along with other mechanisms, to drive the bubbles discovery and management, initiating the creation of new bubbles based on the inferred conclusions. Moreover, it proposes the use of recent security and privacy-aware approaches for the operational stage, which are intended to provide a flexible and lightweight data sharing mechanism within bubbles, while security and privacy concerns are addressed. In particular, for this purpose, we consider the Ciphertext-Policy Attribute-Based Encryption (CP-ABE) cryptographic scheme [6], as well as the use of anonymous authorization credentials based on our previous work [7].

The remainder of the paper is structured as follows. Section 2 provides an overview of some general concepts in this research area. Section 3 delves into the bubble management proposal, including the architecture, the bubble discovery process and operational aspects definition. Section 4 describes a use case as an example of applicability of the proposed solution. Section 5 concludes the paper with some final remarks and future works.

II. Social IoT and Bubble concept

The different types of relationships among smart objects have been already studied in Social Internet of Things [4]. In our work we rely on those kinds of relationships, but putting the emphasis in the common interests that smart objects and their owners may have to set up the dynamic groups. Therefore, we can identify five main types of relationships: a *Personal object relationship* can be established among objects belonging to a same owner; a *Co-location object relationship* is given among objects that are placed in close locations but without needing to be placed always in the same places, i.e. among objects whose distance in a certain moment is lower than one predefined; furthermore, a *Common Interest relationship* can be established among users or smart objects that share the same or similar target goal (e.g. to provide a common IoT application); a *Social object relationship*:could be set up among smart objects because of the social relations among their owners; finally, a *Parental object relationship* is defined among similar devices (e.g. built in the same period by the same manufacturer.

Such relationships drive how different smart objects can be grouped, establishing bubbles in which a specific set of security restrictions must be enforced. In this way, a *Personal Bubble* can be composed of a set of smart objects belonging to the same owner, or a set of cars from the same manufacturer could be grouped comprising a *Parental Bubble*, in order to provide information about the engine condition of the vehicles. Furthermore, such bubbles can be set up in an opportunistic way [8]. In this sense, an *Opportunistic Bubble* can leverage opportunistic contact among smart objects to detect relationships that originate this kind of groups. Opportunistic bubbles could be established spontaneously (e.g. based on physical proximity), and using short-range communications to realize a data sharing mechanism among smart objects of the same bubble. For example, Alice can initiate the creation of an opportunistic bubble of smart objects when her smartphone detects some of her friends are in the same music festival. Indeed, because of the inherently mobile nature of smart objects (such as mobile phones or vehicles), this model has an significant potential to be exploited in IoT. In this work, we propose a mechanism based on clustering and CEP, in order to facility the creation and discovery of such relationship-based opportunistic bubbles. These mechanisms are intended to be integrated within our proposed IoT security framework [9], as detailed in the next section.

III. Bubble Management Proposal

This section is devoted to explain in detail the inner logic of the proposal. Firstly, an overview of the architecture is put forward. Next, the process for creating and discovering bubbles is described. Finally, how such bubbles operate once they have been established is also stated.

A. Architecture

The architecture and functionality of the proposed approach is enclosed as part of the SOCIOTAL EU project[1] framework which, in turn, is based on the Architectural Reference Model (ARM) of IoT-A. Based on the Functional View of ARM, this framework extends the security functional group by defining additional functional components, which are intended to deal with the dynamic, pervasive and distributed nature of IoT scenarios [10]. Specifically, in addition to the five functional components of ARM, this framework proposes an extension with the inclusion of two new functional components: Context Manager and Group Manager. The former is intended to enhance the rest of security components with context awareness features, in order to foster the design of adaptive security mechanisms to be leveraged by smart objects. The latter has the aim of dealing with more flexible and data sharing models, in which a group of entities can be involved, while security and privacy need to be preserved.

Moreover, the proposed security framework defines different interactions among security functional components in order to develop suitable security and privacy-preserving mechanisms for IoT environments. While such framework provides a global overview about the requirements of security and privacy, in IoT, this work focuses on the main interactions required for the design of the proposed mechanisms, involving the Context Manager and the Group Manager. Specifically, under the common view of our IoT security framework, Figure 1 shows how these components are instantiated by a smart object (e.g. a smartphone or actuator), and an infrastructure component, which is intended to be responsible for the creation, management and discovery of relationship-based opportunistic bubbles of smart objects.

B. Bubble Discovery Process

When a Smart Object has local-processing capabilities, it can automatically start the discovery of bubbles when certain changes of its surrounding context, preferences and/or features occur. The underlying idea is that if there is a *shift* of some of these characteristics, the new *state* might make the object suitable for being appended in new or existing bubbles. In addition to that, the infrastructure side can also initiate the discovery of new bubbles on behalf of those objects that do not have enough local processing capabilities. Thus, the whole process for bubble discovery involves the following steps.

1) Bubble-related change detection: The responsibles of detecting the aforementioned changes are either the local rule engine (LRE) of the smart object or the global one in the infrastructure (RE) (see Fig. 1). Since these changes should be detected in a timely manner, both modules are developed

[1] http://sociotal.eu

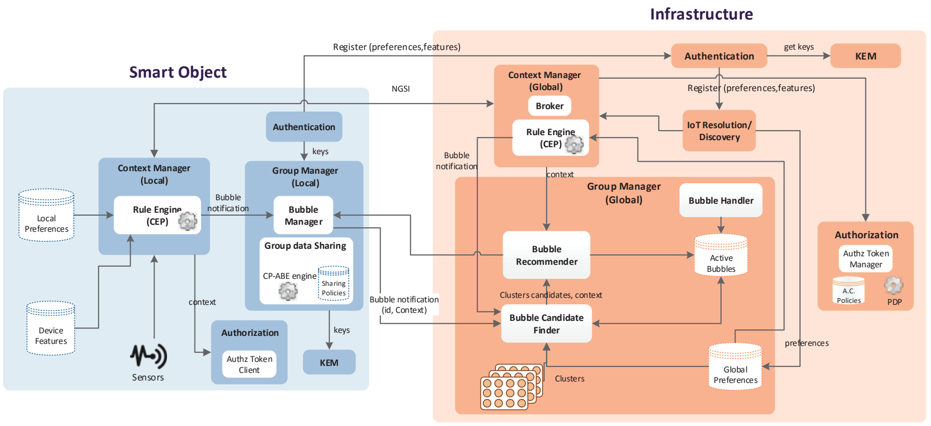

Figure 1. System architecture.

by following the Complex Event Processing (CEP) paradigm [5].

CEP is a software paradigm that intends to capture certain situations of interest in nearly real time. This is carried out by means of pre-defined Event Processing Rules (EPRs). As a result, a CEP system emits a palette of derived events representing the target situations of interest.

The two rule engines are co-located with the local context manager (LCM), in the smart object, and the global one (GCM) in the infrastructure side. Whilst the LCM stores and manages the contextual information relevant for the smart object (coming from internal and external datasources), GCM aggregates the context information from the objects in order to extract a more general contextual knowledge.

Fig. 2 depicts the logic structure of CEP-LRE, comprising four Event Processing Agents (EPAs). An EPA consists of a set of EPRs that deals with similar incoming events and compose related derived events. Thus, each EPA focuses on perceiving changes of particular types of information related to its object.

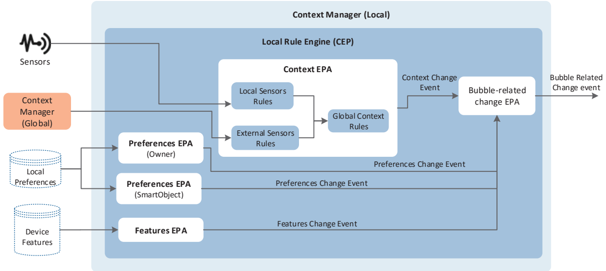

Figure 2. Local Context Manager architecture

To be specific, the *Context EPA* comprises the rules in charge of detecting meaningful changes within the surrounding context of the object. This way, as Fig. 2 shows, three types of rules are enclosed in this agent, *local sensor rules* that detect contextual changes from the object's built-in sensors (e.g. accelerometer, GPS, microphone, etc.), *external context rules* that focus on changes from external contextual sources, namely contextual information from the GCM which is relevant for the smart object, and *global context rules* that aggregate and correlates contextual changes from the two previous types of rules so as to come up with a higher level of perception.

The *Preferences EPA* comprises the rules that would fire when some of the smart objects' local preferences are modified. In that sense, it is possible to distinguish between preferences directly related to the smart object's owner, which are independent of the particular smart object, and the object's ones. For that reason, two types of rules, one for each type of preferences, are included in this EPA (see Fig. 2). In that sense, it should be pointed out that the concrete attributes of both types of preferences are very dependant on the type of smart object and the domain of application.

The *Features EPA* is in charge of detecting new or modified values of the available technical features of the object. For example, this EPA could detect if the owner has activated the built-in GPS sensor as this provides the device with new location capabilities

All the different events emitted by the three previous EPAs are finally processed by the *Bubble-related change EPA* that correlates and aggregates all these events looking for bubble-related changes that could give raise to a bubble discovery process.

For the sake of completeness, the logic structure of the RE in the infrastructure is quite similar than the one depicted in Fig. 2 but with no local sensor rules.

2) Bubble-discovery request: Each time bubble-related changes emitted by the CEP-LRE are received by the LGM, this module checks if the smart object can take part of a new bubble or not depending on several factors like the preferences and the technical features of the object.

In case the module decides to start a new discovery process, a *bubble discovery notification* is sent to the infrastructure (see Fig. 1). This notification includes the identifier of the smart object. Moreover, if it was originally caused by a context change of the object, such contextual information is also appended to the notification.

3) Bubble Candidates Search: Once the infrastructure receives a *bubble discovery notification*, such message is processed by the *Bubble Candidate Finder* (BCF) (see Fig. 1) which makes up a set of candidate bubbles the requester object could belong. In order to generate such candidates, the BCF composes pre-defined groups of smart objects with similar preferences and compatible technical features. This is possible as the global preferences repository stores the preferences and features of all the target smart objects.

In order to compose such aggregations, the present work proposes a fuzzy clustering approach [11]. Basically, a fuzzy clustering algorithm intends to divide a dataset into fuzzy partitions or *clusters* comprising elements with certain similarities. Unlike hard partitions, elements belong to clusters with a certain *membership* degree.

Consequently, the BCF applies a fuzzy clustering algorithm to the data in the global preferences repository. To do so, the data in such a repository is regarded as a set of tuples, each one representing a particular smart object. This way, the tuple of a smart object s with owner *o* takes the general form of s:$\{o_{pref}, s_{pref}, tech_{feat}\}$. In that sense, a *numerization* of the nominal fields of the tuples is required in order the algorithm to perform the clustering. As a result, a set of clusters of tuples representing groups of objects with similar preferences is generated[2].

Nevertheless, as it was stated in section III-A, each smart object defines a set of technical preferences or needs for other objects to share the same bubble. These needs will vary from one object to other. Consequently, a different clustering process should be carried out for each smart object by only considering those ones accomplishing its preferences.

In addition to that, since the content of the global preferences repository will tend to change throughout time, the aforementioned clustering processing will be re-launched whenever a certain *disparity* criteria between the current clusters and the most recent data is accomplished.

For the sake of clarity, Algorithm 1 shows the pseudocode of the whole process of candidate bubble generation representing the selection of smart objects accomplishing the needs of a particular one (lines 4-6) and the generation of the candidate bubbles for such object (line 7).

All in all, when a new *bubble discovery notification* from a smart object s is received, the BCF takes its most updated set of candidate bubbles CB_s and delivers such information to the *Bubble Recommender* (BR).

4) Final bubbles recommendation: The BR module is in charge of deciding which of the candidate bubbles from the BCF are eventually composed by means of a two-step procedure. The pseudo-code of this decision process is shown in Algorithm 2.

Algorithm 1: Composition of candidate bubbles.

```
   Input: Set of smart object tuples $SO$
   Output: Set $CB$ of candidate bubbles $CB_s$ of each
           smart object $s$
1  for each $s \in SO$ do
2      $SO_s \leftarrow \emptyset$
3      for each $s_{aux} \in SO$ do
4          if $s_{aux}.tech_{feat}.accomplish(s.s_{pref})$ then
5              $t \leftarrow$ numerization($s_{aux}.o_{pref}, s_{aux}.s_{pref}$)
6              $SO_s \leftarrow SO_s \cup t$
7      $CB_s \leftarrow$ perform_clustering($SO_s$)
8      $CB \leftarrow CB \cup CB_s$
9  return $CB$
```

Algorithm 2: Generation of final bubbles.

```
    Input: Smart object $s$ initiating the bubble discovery,
           Set $CB_s$ of candidate bubbles of smart object $s$,
           Set $AB$ of active bubbles in the deployment
    Output: Set $FB_s$ of final bubbles for smart object $s$
1   $FB_s \leftarrow \emptyset$
2   for each $cb \in CB_s$ do
3       $fb \leftarrow \emptyset$
4       for each $s_{cb} \in cb$ do
5           if $s_{cb}$.compatible_with(s.dyn_context) then
6               $fb \leftarrow fb \cup s_{cb}$
7       if $fb \not\subset AB$ then
8           $FB_s \leftarrow FB_s \cup fb$
9   $AB \leftarrow AB \cup FB_s$
10  return $FB_s$
```

Firstly, for each candidate bubble *cb*, the module checks if its smart-objects (s_{cb}) are *compatible* with the dynamic context of the object that originally initiated the bubble discovery (s) (lines 6-8) of Algorithm 2. For each $< s, s_{cb} >$ pair, this compatibility means that either both objects share the same context (e.g. they are located close to each other) or at least their context do not interfere with each other. Thus, only compatible objects s_{cb} are retained for each final bubble fb.

Secondly, in order to avoid duplicities, only bubbles *fb* that are not already active ones are included in the final set FB_s (lines 7-8 of Algorithm 2). The resulting set is considered the new active bubbles generated by the process (line 9 of Algorithm 2).

In the end, for each bubble in FB_s, the BR informs its

[2] Note that, due to the fuzziness of approach, a smart object can belong to more than one cluster at the same time.

member about their inclusion in the bubble which. In the smart object side, this notification is processed by the LGM (see Fig. 1).

C. Bubble Operational Process

After a group of smart objects is notified (or they have discovered) a bubble, such devices enter the operational stage exchanging information related to the purpose of the created coalition. During this stage, the application of security and privacy-preserving mechanisms is crucial to ensure a proper and effective operation of the bubble, in order to avoid any data leakage out the bubble. Towards this end, our approach is based on the use of two main mechanisms: the CP-ABE cryptographic scheme [6], and anonymous authorization credentials [7].

The use of CP-ABE is intended to allow information to be shared among members of a bubble, enabling secure one-to-many communication. This scheme can be applied, for example, to allow information sharing through the well-known publish/subscribe pattern through the infrastructure component, in which members of a bubble (acting as subscribers) are able to decrypt information being disseminated by other members (acting as publishers). Through the use of CP-ABE, a piece of data can be encrypted under a policy of attributes, while keys of participants are associated with sets of attributes. Thus, we assume that, when a smart object is registered in the infrastructure component, it receives a CP-ABE key associated to the $tech_{feat}$ set, as well as the set of public parameters that are required to encrypt/decrypt information. Since the set of preferences is associated with the technical features of the smart object, unlike traditional of symmetric cryptography schemes, the same CP-ABE key can be used to participate in different bubbles without requiring the generation and delivery of new key. Thus, once the bubble is created, the *Group data Sharing* module (see Figure 1) will be responsible for disseminating the information so that only those members of the bubble can access it.

The use of anonymous authorization credentials is intended to enable M2M communications among members of the same bubble, via a flexible and lightweight approach. These credentials could contain the set of privileges for a specific device within a particular bubble. These privileges will be determined by the information associated to the smart object itself (i.e. their technical features), as well as their preferences, which are registered in the infrastructure component previously. Thus, when a new bubble is discovered and notified to a smart object, this notification message will contain a credential associated with a specific bubble (indicated in the ”de” field), and the list of access rights (ar field) available to the smart object in such bubble. It should be pointed out both mechanisms are complementary and can be used for secure 1-to-many and 1-to-1 communications, respectively.

IV. Use case - Opportunistic Study Groups

We describe here how the proposed method can be used to create opportunistic study groups bringing together students with common subjects of interest, promote unplanned meetings of students, teachers and/or other educational staff allowing them to study and discuss about their common interests at the same time their associated devices automatically exchange information about such topics and the room's elements where they are located provide a suitable environment for knowledge sharing, like showing associate content in the room's displays or setting a suitable temperature.

In this scope, each study group will be represented as an opportunistic bubble comprising smart objects of both people (e.g. laptops, smartphones, etc.) and the institution's facilities such students belong to (e.g. displays, temperature controllers, etc.). The general guidelines to come up with such type of bubbles are devised next.

1) Bubble-related change detection: Since a study group needs all its members to be together in the same space, a student's smart object could initiate the discovery of new bubbles (study groups) when it detects a meaningful spatial movement. For that goal, an event-based rule is defined in the *Context EPA* within the CEP-LRE of each student's smart objects that detects when the object enters (and, thus, its owner) a new building,

```
CONDITION GPSLocation l1 ->
          GPSLocation l2
          AND closestBuilding(l1)≠
          closestBuilding(l2)
ACTION    new NewBuidingEvent(
            closestBuilding(l2))
```

In particular, the rule detects when the user has moved from location `l1` close to a facility to a new location `l2` close to a different facility. Both locations are measurements from the GPS sensor of the object. As a result, a *new building event* representing the new facility is created.

Next, the *bubble-related change EPA* correlates each *new building event* with the last *preference change events* so as to ensure that the object's owner allows to take part of bubbles when he is in the new facility. If that is the case, a new *buble-related change event* is delivered to the LGM.

2) Bubble-discovery request: For each new *buble-related change event*, the LGM checks that its holder object does not already take part of an active study group (bubble) due to the fact that a person can not be at two different meetings at the same time. If no active bubbles are registered, a *bubble notification* is sent to the central server.

3) Bubble Candidates Search: In order to make up the fuzzy clusters of smart objects acting as candidate bubbles, we should consider that, in this use case, two types of smart objects exist, the personal ones of students, teachers, etc. and the ones which are part of the facilities' infrastructure. Since both of them have quite different characteristics, two different clustering processes (Algorithm 1) are launched, one for the personal objects and other for the infrastructure ones.

Regarding the clustering of personal objects, the BCF in the infrastructure considers, for each smart object s, a set of preferences of its owner to take part of study groups o_{pref}={*degree, grade, subjects of interest, educational role*} along with ones related to the object itself s_{pref}={*data-sharing type,* $tech_{feat}$}. In this case, $tech_{feat}$ refers to technical features of other objects required to interact with s (e.g. specific manufacturer, etc.). As a result, the candidate bubbles of personal objects will group together those objects whose owners have similar preferences in terms of study groups and, also, compatible data sharing preferences and technical features.

Concerning an infrastructure object, its preferences mainly have to do with the policy of its host facility concerning hours of operation. Also, since a infrastructure object usually has a fixed location, such a feature is also considered in the clustering. This way, the candidate bubbles of infrastructure sensors represent groups of bubble located in the same space (e.g. office, classroom, etc.) with similar active hours.

Therefore, each time a new *bubble notification* from a smart objects s is received by the central server, the BCF selects two types of candidate bubbles, 1) the set CB_s^{per} of personal smart objects with similar preferences and compatible with s in terms of technical features and 2)CB^{inf} of infrastructure smart objects. Then, each pair in $CB_s^{per} \times CB^{inf}$ is merged giving raise to the complete set of candidate bubbles CB_s.

4) Final Bubbles Recomendation: Lastly, the BC module launches Algorithm 2 with CB_s. In this case, the *compatibility* function checks, for each candidate bubble cb that 1) all its objects that include location as a dynamic context property are actually in the same facility, 2) the current hour of the day is compatible with the operational hour of its infrastructure sensors and 3) those infrastructure sensors are in the same facility than the personal smart objects. Note that this allows personal smart objects that do not specify location as a dynamic contextual feature (like desktop computers) to belong bubbles.

Finally, for each new discovered bubble, the owners of its personal smart objects are notified that a new study group has been composed along with its topic and location.

V. Conclusions

The IoT and Big Data paradigms already demand more self-managing models to allow smart objects to proactively establish trust relationships among them.

In this work we devised the general guidelines so as to come up with opportunistic groups of smart objects (or *bubbles*) by following a CEP and fuzzy clustering procedure. Whilst CEP allows to detect changes in the state of a smart object to timely initiate the discovery of new bubbles, the fuzzy clustering is used to define *prototypes* of bubbles that are refined later with dynamic contextual information.

Further work will focus on following the guidelines devised here in order to develop bubble management frameworks in real scenarios.

Acknowledgements

This work has been sponsored by European Commission through ENTROPY-649849 EU Project and also by the Spanish MICINN, project EDISON (TIN2014-52099-R).

References

[1] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, “Smart cities and the future internet: Towards cooperation frameworks for open innovation.” *Future Internet Assembly*, vol. 6656, pp. 431–446, 2011.

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

[4] 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*, 2012.

[5] O. Etzion and P. Niblett, *Event Processing in Action*. Manning Publications, 2010.

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

[7] J. L. Hernández-Ramos, J. B. Bernabe, M. Moreno, and A. F. Skarmeta, “Preserving smart objects privacy through anonymous and accountable access control for a m2m-enabled internet of things,” *Sensors*, vol. 15, no. 7, pp. 15 611–15 639, 2015.

[8] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, “Opportunistic iot: Exploring the harmonious interaction between human and the internet of things,” *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1531–1539, 2013.

[9] J. B. Bernabe, J. L. Hernández, M. V. Moreno, and A. F. S. Gomez, “Privacy-preserving security framework for a social-aware internet of things,” in *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*. Springer, 2014, pp. 408–415.

[10] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.

[11] R. Babuška, *Fuzzy modeling for control*. Springer Science & Business Media, 2012, vol. 12.