

# Securing the Web of Things with Role-Based Access Control

Ezedine Barka<sup>(✉)</sup>, Sujith Samuel Mathew, and Yacine Atif

College of IT, UAE University, Al Ain, UAE  
ebarka@uaeu.ac.ae

**Abstract.** Real-world things are increasingly becoming fully qualified members of the Web. From, pacemakers and medical records to children's toys and sneakers, things are connected over the Web and publish information that is available for the whole world to see. It is crucial that there is secure access to this Web of Things (WoT) and to the related information published by things on the Web. In this paper, we introduce an architecture that encompasses Web-enabled things in a secure and scalable manner. Our architecture utilizes the features of the well-known role-based access control (RBAC) to specify the access control policies to the WoT, and we use cryptographic keys to enforce such policies. This approach enables prescribers to WoT services to control who can access what things and how access can continue or should terminate, thereby enabling privacy and security of large amount of data that these things are poised to flood the future Web with.

**Keywords:** Web of Things · Privacy · Access Control · RBAC · UCON

## 1 Introduction

Today society is impacted by revolutionary innovations in information technology that are very pervasive and ubiquitous in nature. Along with these advances, particularly in communications technology, a series of new security threats and privacy issues arise. Among these technologies is the rapidly increasing Web of Things (WoT), where physical things are accessed and controlled via the Web. WoT has several methods that support a variety of applications such as subscribing to a service, notification of an event, status update, and location and presence services. WoT provides flexible, scalable, and real-time communications with the physical world in a ubiquitous way but additional security and privacy concerns result from its ubiquity and mobility.

Secure Web publishing approaches have been developed to allow authenticated users direct access to a dataset. In doing so, these approaches provide users with a published, static “snapshot” of the dataset content. We follow this secure publishing paradigm [5] to enable a security framework for WoT.

Traditional access controls typically focus on the protection of data in closed environments, and the enforcement of control has been primarily based on identity and attributes of a known user. These types of access control lack a comprehensive, systematic approach to fulfill the security requirements of today's

pervasive and ubiquitous applications on the WoT. To address these issues, we introduce an architecture that implements role-based access control (RBAC) to check the access to datasets within WoT based environment. This enables publishers of things on the Web to control who can locate them, and subsequently access and use them. Furthermore, it enables the possibility of setting some attributes to determine whether certain accesses should proceed or be terminated.

The remainder of this paper is organized as follows. In Section 2, we provide some background on WoT and discuss our architecture and its role in the pervasive environment to address some security challenges. Section 3 provides an overview of the role-based access control (RBAC). Section 4 presents our architecture and explains the integration of WoT with RBAC. Section 5 describes how RBAC is used to specify the access policies to WoT datasets, and the cryptographic keys used to enforce these policies. Section 6 concludes the paper with some future work.

## 2 Overview of WoT

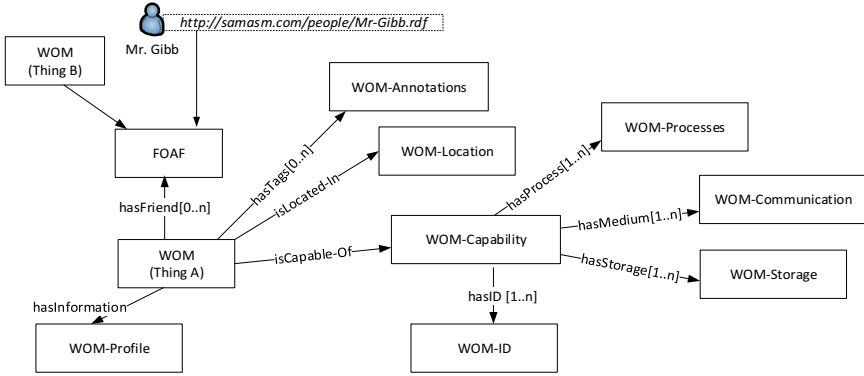
WoT is a platform where billions of physical things are interconnected over the World Wide Web. Researchers have successfully connected things over the Web and experimented with various applications in real-world scenarios [4]. The inevitable challenges lie in how to efficiently and effectively manage and secure the access to the information hidden within these things, which is critical for a number of important applications. To address the management of heterogeneous and wide abundance of candidate things in WoT, the Ambient Space Manager (ASM) framework was suggested earlier by Mathew et. al [10]

### 2.1 Representation of Things on WoT

Mathew et. al. suggested a capability based classification, Fig 1 shows the Web Object Metadata (WOM) structure, which defines the ontological representation of a thing (Thing A) on the Web [6].

The *WOM-Profile* composes the semantic details from all ontologies of a thing that is revealed to external entities. The WOM-Profile is divided into two sections: the *ipreset* and *idynamic* sections. Preset describes static information about a thing like manufacturer, date of production, or country of production and the dynamic, describes information about a thing like cost, location, or owner, which changes. The WOM-Capability ontology classifies a thing based on its Identity, Processing, Communication, and Storage (IPCS) capabilities. The ontology classifies a thing to be *Web Smart* when these capabilities are Web related. Hence a Web Smart thing has a unique identity on the Web, processes Web requests, communicates via Web protocols, and has storage space on the Web. If any of the capabilities are missing, then the ontology recommends the augmentation of the missing capabilities.

Once things are Web Smart (i.e. they are participating members of the Web), they are grouped/clustered into an Ambient Spaces (AS) [10,9]. An AS is the



**Fig. 1.** Web Object Metadata (WOM) of a thing on the Web

virtual representation of a cluster of things i.e. the encapsulation of one or more real-world things that are Web Smart. An AS also represents the boundaries of a physical space. For example, Web Smart things in a classroom, or in a train compartment, or a hospital room, or a parking spot. These physical spaces are repeating patterns. Hence an AS provides a template to compose things and their containing physical spaces in a gradient to represent larger physical spaces like campuses, parking lots, airports, trains, and office buildings. Clustering things into an AS is done based on determining the similarities of things using similarity functions. The similarity functions are applied on all Web Smart things in an AS [8].

## 2.2 Ambient Space Stakeholders

In any fundamental computing setup, the main stakeholders are the providers and consumers of the services or infrastructure. The consumers use and update the system, while the providers deal with the manufacture, deployment and maintenance functions. The domain of WoT requires the addition of new stakeholders and redefinition of the traditional ones. The stakeholders within the WoT domain not only require providers and consumers but also needs to consider the role of owners and regulators who control the thing's inherent dynamic and proprietary state. Here, we briefly list the stakeholders, focusing on their contribution to the content of a thing's WOM-Profile.

**Providers:** The providers are essentially the manufacturers that create the WoT elements. The providers will also hold the responsibility of recycling or discarding a thing at the end of its lifespan [7]. The maintenance and upgrades to a thing are the responsibility of providers while a thing is used by other stakeholders. The providers hold the right to change the content of a thing while maintaining history of changes. The providers contribute to the preset content `!wom:preset!` of a thing's profile and are responsible for ensuring the presentation of thing's composition, use, and disposal. The preset content of a thing's

WOM-Profile is fixed and not changeable by other actors. Contact information of the providers needs to be provided, for the use of thing itself or any of the other stakeholders. The links to the user manual and the conditions of thing's usage are provided by the providers. The providers may also contribute to the dynamic content `⌈wom:dynamic⌋` of a thing's prole. Annotations for branding, price composition and marketing are initially added by the providers. The providers initiate the history of a thing's existence.

**Consumers:** The consumers of a Web Smart thing are its users. These users could be other *things* or people. Unlike other domains, consumers are not owners here and are bound to access restrictions that are controlled by the present owner of a thing. The contribution of consumers populates the dynamic content `⌈wom:dynamic⌋` of thing's prole. The consumers provide rich semantics to thing's use and add to the history of a thing. The content that the consumers provide to a thing essentially creates links with other *things* or people that are connected to the consumer. Thus the consumers play an important role in promoting *thing's* social connectivity.

**Owners:** The owners are consumers but have more rights to a thing's usage and content. The owners provide access restriction to a thing's operations and can loan or lease a thing. With proper authorization from regulators and providers, the owners can alter the dynamic content `⌈wom:dynamic⌋` of a thing and therefore change history. The options to re-brand or marketing a thing allows owners to change the value of a thing and promote its acceptance among other *things* or people.

**Regulators:** While the other stakeholders provide content to value a thing, the role of the regulators prevails over other stakeholders. For example, government authorities or regulatory authorities that ensures the safe, sustainable, and judicious use of Web Smart *things*. The regulators provide details on rights and obligations of other stakeholders. They provide contractual details wherein other stakeholders and authorities are informed if there is a breach of contract. Because of the wide spread implication of the virtual use of physical things, liabilities and exceptions are to be clearly defined by regulators. For the WOM-Profile, the regulators provide content that are both preset and dynamic related to issues like privacy, trust, cyber-attack and legal implications. The role of regulators needs to be actively researched, investigated, and formulated with government and international bodies so as to ensure the secure and sustainable use of *things* on the Web.

Manufacturers follow a structured product labeling standard to provide consumers with the information of a thing's content and usage. The process of monitoring and regulating these standards become easier when the information is digitally embeded or appended to products. The benefit of using the WOM-Profile as a digital standard for communicating product infomration is two-fold. Firstly the standard information can be included in the `⌈wom:preset⌋` part and secondly user experiences can be included in the `⌈wom:dynamic⌋` part of the WOM-Profile. While it is important to understand the semantic structure of

Web Smart thing's information and the major stakeholders, it is also important to realize how the information is stored and retrieved from real-world things.

### 2.3 WoT Framework

The AS enables real-world things to be imbibed into the WoT ensuring seamless communication between people and things. This opens up many social applications that is bound to enhance business and industry. Some applications were suggested based on the ASM framework [5,6]. Here, we take an example of how classrooms are virtually represented as Ambient Spaces, to describe the framework. Fig 2, depicts each classroom in a school campus as an Ambient Space (AS).

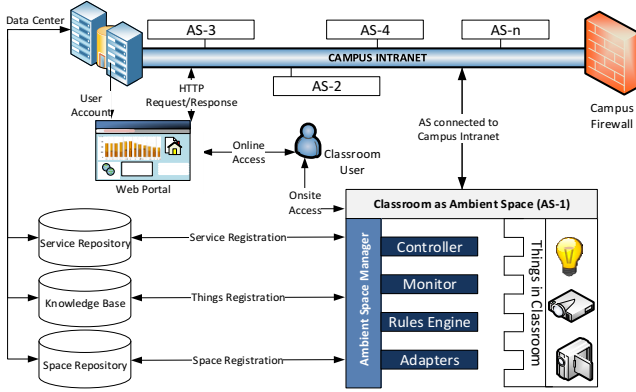


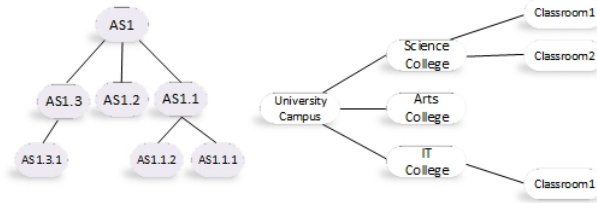
Fig. 2. Subsuming classrooms into the WoT using Ambient Spaces

Each AS is controlled by an Ambient Space Manager (ASM) which includes the Controller, Monitor, Rules Engine, and Adapters. These modules provide essential management functionalities that provide the access and control of things in an AS. The Service repository, Knowledge Base, and Space repository contain the information that is relevant to all AS. The users has both onsite and online access to things in an AS.

The ASM framework creates a hierarchical structure for representing physical spaces and the things therein. Fig 3, provides a general depiction of the structure and also an example. Similar structure is suggestive to represent hospital rooms, train compartments, seats on an international flight, or in a movie theatre. Thus the ASM framework provides a scalable structure to represent physical things on the Web and populate the WoT.

### 2.4 WoT Security Challenges

Openness and sharing are always contradictory when it comes to security and privacy. A practical consideration for enabling widespread adoption of WoT is



**Fig. 3.** Representing repeating patterns of physical spaces and things in them with Ambient Spaces

the security and privacy vulnerabilities of shared resources of things and related data. Moreover, how does the framework verify Web services and estimate their reliability against malicious intervention or inadvertent errors. Although security solutions and related technologies have been developed to protect systems against many vulnerabilities, most of these technologies do not have a cohesive structure to deal with the security issues specifically related to the WoT, and advocate ad-hoc approaches instead. This is because WoT introduces new dimensions of risk, due to its heterogeneous and ubiquitous nature. Some of the threats that are inherent to the use of WoT are listed as follows:

- Impersonating a server: A WoT user contacts a Proxy server to deliver requests. The server could be impersonated by an attacker. The mobility of things further complicates this scenario.
- Tampering with message bodies that contain requests.
- Tearing down sessions – insert a disconnect command.
- Denial of Service attacks - Denial of service attacks focus on rendering a thing on the Web unavailable, usually by directing an excessive amount of network traffic to its interfaces. The WoT face the public Internet in order to accept requests from worldwide IP endpoints, which creates a number of potential opportunities for distributed denial of service attacks that must be recognized and addressed by the implementers and operators of this ecosystem.

Therefore, the security challenges facing WoT is to ensure the following:

- Data Security and Privacy: How to protect the thing’s data and private information and locations? In WoT, addressing the issue of data security is particularly challenging, due to the unique features of the network, such as mobility of the entities and the size of the network. It is essential that thing’s critical information is protected from being inserted or modified by attackers. For privacy, the challenge is on how to ensure a conditional privacy in the sense that thing’s private information like identity, speed, or location are protected from unauthorized access while access should always be granted when needed by authorities.
- Authentication: Most technologies use Web services today and have the HTTP style access mechanism which is not foolproof when dealing with

real-world things. A single sign-on authentication mechanism is at-least required.

- Authorization using policy-based mechanisms: The Read/Write/Execute controls that are embedded in file systems. Earlier recommendations have tried implement, traditional access control models, but they are broadly categorized as discretionary access control (DAC) [3,12] and mandatory access control (MAC) models [3,12]. Others have proposed new models such as role-based access control (RBAC) and task-based access control (TBAC) to address thee security requirements [13,16].

None of the above mentioned solutions are sufficient in isolation for providing security for a large-scale, distributed and sometimes resource constrained pervasive environment like in WoT context. Hence, our approach utilizes the well-known Role-Based Access Control (RBAC) to control access to things on the Web.

There are many benefits to adapting RBAC to WoT context. RBAC supports data abstractions which enables subscribers to WoT services to control who can identify the locations of the things, to approve or disapprove subsequent access, and to also set parameters to determine whether a certain accessn can continue or should terminate. RBAC also enforces other security concepts that are specific to some applications such as lease privileges or separation of privileges. In this case, RBAC may deny the access or connection when the requested authorization of the prescriber does not meet the access control policy requirement or the thing's attribute changes.

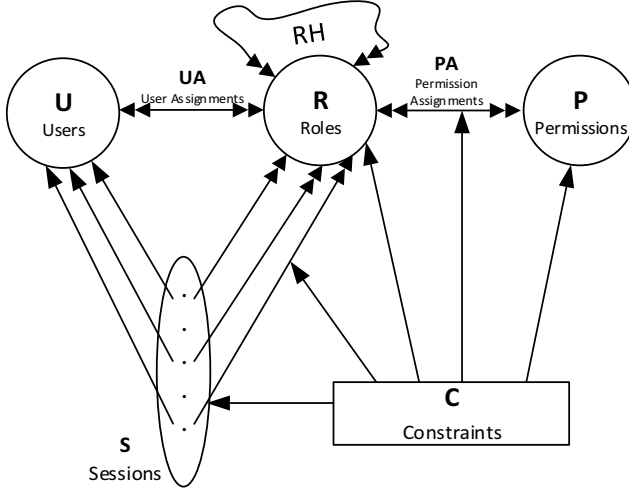
However, RBAC is susceptible to role proliferation. For example, thousands of users may be granted access to various parts of a thing's dataset. The access permission my differ depending upon each user's affiliation with the system. This scenario my demand that role-based policy assigns one role to each user, which can be too much to handle. Therefore, the concept of role parameterization, developed by [3], has shown to be an effective way to deal with the issue of role proliferation. The following section provides an overview of the RBAC model.

### 3 Overview of Role Based Access Control (RBAC) Model

In this section we briefly review the general ideas of RBAC and the core authorization models. The details of these models can be found in [2,14,1].

RBAC is proven to be a good alternative to traditional discretionary and mandatory access controls. It ensures that access to certain data or resources is given to authorized users only [14]. It also supports some important security principles such as least privilege, separation of duties, and data abstraction. Least privilege is supported, because RBAC is configurable such that only those permissions are assigned to the role required for the tasks conducted by members of the role. Separation of duties is achieved by ensuring that mutually exclusive roles must be invoked to complete a sensitive task, such as requiring an accounting clerk and account manager to participate in issuing a check. Data abstraction is supported by means of abstract permissions. Instead of the read, write, and

execute permissions typically provided by the operating system. Other permissions such as join, leave, join as a sender, or join as a receiver, are also be expressable.



**Fig. 4.** Basic RBAC Model

A general RBAC model was defined by Sandhu [14] and is summarized in Fig 4. The model is based on three sets of entities called users (U), roles (R), and permissions (P). A user is a human being (an entity that seeks access). A role is a function with some associated semantics regarding the authority and responsibility conferred on a member of the role. Permission is an approval of a particular mode of access to one or more users in the system. The user assignment (UA) and permission assignment (PA) relations of Fig 4 are both many-to-many relationships (indicated by the double-headed arrows). A user can be a member of many roles, and a role can be assigned to many users. Similarly, a role can have many permissions, and the same permission can be assigned to different roles.

Role hierarchy (RH) in RBAC is a natural way of organizing roles to reflect the lines of authority and responsibility. The hierarchy is partially ordered, so it is reflexive, transitive, and anti-symmetric. Inheritance is reflexive because a role inherits its own permissions. Transitivity is a natural requirement in this context, and anti-symmetry rules out roles that inherit from one another and are therefore redundant.

## 4 Security Architecture for WoT

Integrating the RBAC technology into ubiquitous WoT-based environment requires a careful mapping between the entities of RBAC and those entities



and components of the WoT. Following is a list of integrated components which require such mapping:

- User/Subjects: The concept of participants in WoT is represented as a user component in the RBAC.
- Permissions/Rights: The concept of permissions in RBAC is captured through the privileges that a WoT participant needs in order to complete a task.
- Objects: the concept of objects in RBAC are used to represent all resources *things* that a WoT participant seeks to access or to connects to.
- Authorization Rules: Authorization rules in RBAC are the set of requirements that should be satisfied before any WoT user be permitted to establish any connection with, or to access any other WoT entity.
- Session: The concept of session in RBAC is captured in WoT by the set of durations for which WoT entities are active.

#### 4.1 Integrating RBAC in WoT

One of the most critical issues in using RBAC for enforcing the specified access policies in WoT environment is to use the concept of a reference monitor (RM), which has been introduced, and extensively discussed by the access control community for years, and has become the ISO standard for access control framework [15].

The RM concept has been considered as the core control mechanism for access and usage of digital information. In classical access control, subjects access digital objects only through the reference monitor, which is a process inside the trusted computer base that is always running and is a tamper proof.

The following section discusses our conceptual structure of RBAC/WoT access control domains, based on the reference monitor.

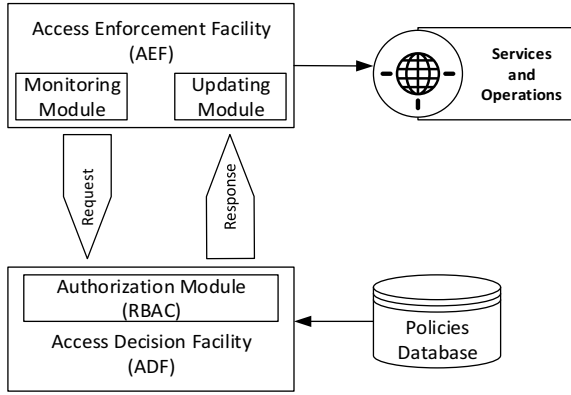
#### 4.2 Policy Enforcement Facilities

In our architecture, we use a customized version of the well-known ISO reference monitor standard [9].

According to this ISO standard, the reference monitor consists of two facilities: Access Control Enforcement Facility (AEF) and Access Decision Facility (ADF). The AEF and ADF interact with each other in such way that every request by a subject to access an object in the system get intercepted by AEF. The AEF in turn asks the ADF for a decision on whether to approve or disapprove the request, and subsequently the ADF returns either ‘yes’ or ‘no’ as appropriate. The enforcement of this decision takes place at the AEF.

In our architecture, the reference monitor is similar but differs in the details from that of ISO reference monitor. We incorporate the role-based access control to handle the “pre-decision” authorization rule. Fig 5 shows the conceptual structure of the RBAC/WoT reference monitor.

As the Fig 5 shows, any request to access any WoT resource “thing” is intercepted by the AEF. Before making any decisions, the AEF forwards the request



**Fig. 5.** Conceptual Structure for RBAC/WoT Reference Monitor

to the ADF, which in turn adheres to the RBAC policy decision of whether to grant or reject the authorization request. RBAC will allow authorization of an active (subject) entity to execute a certain right on a passive (resource) entity only if the subject belongs to a role that RBAC has previously assign that right to.

The rest of the decision process by AEF would continue only if RBAC grants authorization, otherwise the process is stopped and response by ADF is negative (no authorization). Furthermore, RBAC allows authorization after it tests other decision factors, mainly, hierarchal relationships and constraints. For example, if the condition for granting authorization is met (i.e., the request is within the range of the allowed operating time), and also the requester agrees to accept to perform a certain obligation, then the ADF returns a positive response “Authorize” to the AEF, otherwise request is denied.

### 4.3 Areas of Control Architecture

To control the access to the WoT environment, our architecture considers one area of control, based on the location of the reference monitor, which is located at the space manager. We refer to this set up as the server side control domain (SCD), because this is the area where the reference monitor is located and where the access policy to the system resources (things) is enforced. Fig 6 below depicts this architecture.

Fig 6 shows that the control of subject’s access to objects is done centrally. In this setup, the subject can either be located within the network or outside, and the objects may or may not be stored in the client’s storage, depending upon the criticality and sensitivity of the content of the object. If it is not that sensitive, then it can be allowed to reside outside of the server-side storage. However, if the content is very critical or very sensitive, the object must stay within the server-side storage.

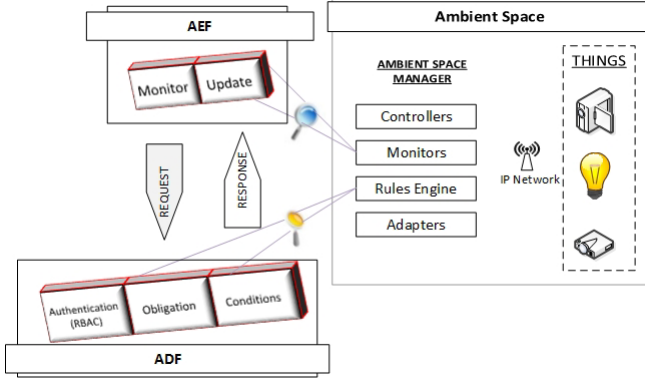


Fig. 6. Integrating RM into SM

## 5 WOT Resources Protection

In this section, we reveal details of the RBAC process for protecting WOM-Profiles of things on the Web. To do this, we adopt the method described by Muldner, Mizilek and Leighton [11]. In this paper, RBAC specifies rules that consist of pairs of the form (role, resources), where a resource is a document fragment specified using an XPath expression (XPath, 2008). RBAC's data abstraction feature allows us to consider any permission needed to control access to the different fragments of an XML document.

### 5.1 Documents and Views

In this paper, access rights are defined using Access Control Policies (ACPs). In other words, ACPs are defined for fragments of XML documents, which we refer to as views. Each WoT activity is published as a single XML document.

Views are specified using a subset of XPath expression referred to as document paths as follows:

**Definition 1.** *A local document path is a document path with no free variables. A free variables are those variables that represent systems variable and their names start with \$. A global document path is a document path which is not local, and considered instantiated when each occurrence of free variables is replaced by some value. For a document  $D$ ,  $P_{D,loc}$  denotes the set of local paths in  $D$ . Each local document path defines a fragment of the document  $D$ . Similarly,  $P_{D,glob}$  denotes the set of global paths in  $D$ . Hence, the set of all document paths is denoted by  $PD = P_{D,loc} \cup P_{D,glob}$ . RBAC is susceptible to role proliferation. Parameterization has been used in the literature to address this problem (REF), and is out of the scope of this paper.*

**Definition 2.** Let  $\Delta\tau$  denotes the language for all roles then, for a WOM,  $D$ , and a finite set of simples roles  $\psi \subset \Delta\tau$  the document-level ACP is a mapping  $\Pi_D : \psi \rightarrow P_D$  such that  $\Pi_D(\psi)$  covers the set  $D$ ; i.e. each element of  $D$  belongs to at least one document path that occurs in the policy. Often, the  $\Pi_D$  mapping is tabulated and shown as tuple  $[(R1, P1), (R2, P2), \dots, (Rm, Pn)]$ .

For a simple role  $R \in \psi$ , if  $\Pi_D(R)$  is local document then it defines a view of  $D$ . If  $\Pi_D(\psi)$  is global document path that contains free variables, then once path is instantiated, it defines a view of  $D$ . The designer of the RBAC policy for, WOM  $D$  may elect to leave some parts of  $D$  unencrypted or make them inaccessible to all users.

For a WOM  $D$ , a finite set of roles  $\psi \subset \Delta\tau$ , and the document-level ACP  $\Pi_D : \psi \rightarrow P_D$  a user in role  $R$  can access precisely the set  $\Pi_D(R)$  and those nodes in  $D$  which are not covered by any path.

## 5.2 Key Generation and Encryption

Let  $\kappa$  be a finite set of keys, where each key is a tuple made of [key name, symmetric key], and  $\kappa_D, \Pi_D$  denotes a document-level key ring for the WOM  $D$  and  $D$ 's policy  $\Pi_D$ , then the key generation for a document-level policy ACP  $\Pi_D : \psi \rightarrow P_D$  takes place as following: If the all paths are local, then each path can uniquely identify a fragment of  $D$ . However, if the paths are global, the issues of parameterization will complicate the case because condition of the path cannot evaluated before the values of the variables are known. For simplicity, we will consider only the local paths.

In this case, a key ring  $\kappa_D, \Pi_D$  is defined and for each  $R \in \psi$ , this key ring defines a set  $\kappa_D, \Pi_D(R)$  of R-Accessible keys. A user in role  $R$  will be provided with R-Accessible keys allowing the decryption of the view  $\Pi_D(R)$ .

To decrypt the document, a user  $U$  will travers the document and use the names of the keys from  $\kappa_D, \Pi_D(R)$  to extract the appropriate key to decrypt the accessible document.

To obtain a key ring that can be used to decrypt a fragment of an encrypted document, a user can request that key ring form the list of roles that user is a member of. Verification of membership can be achieved through presenting the certificate that user obtained membership to that role.

## 6 Conclusion and Future Work

In this paper we introduced a new architecture that encompasses WoT in a secure and scalable manner. Our architecture integrated the features of the well-known role-based access control (RBAC) to specify the access control policies to the WoT. More specifically, we showed how RBAC can be integrated to the WoT architecture to specify access control to the things, which are represented on the Web. We also showed how cryptographic keys are generated and used to enforce such access control policies for these documents. This enable prescribers of WoT services to control who can access their things and how, thereby enables

privacy and the security of large amount of data that these things flood the Web with. Our future work will focus on implementing this architecture.

## References

1. Ferraiolo, D., Cugini, J., Kuhn, D.R.: Role-based access control (RBAC): Features and motivations. In: Proceedings of 11th Annual Computer Security Application Conference, pp. 241–248 (1995)
2. Ferraiolo, D., Kuhn, D.R., Chandramouli, R.: Role-based access control. Artech House (2003)
3. Ferraiolo, D., Kuhn, D.R.: Role-based access controls, arXiv preprint arXiv:0903.2171 (2009)
4. Guinard, D., Trifa, V.: Towards the web of things: Web mashups for embedded devices, Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009). In: Proceedings of WWW (International World Wide Web Conferences), Madrid, Spain (2009)
5. Mathew, S.S., Atif, Y., Sheng, Q.Z., Maamar, Z.: Towards an Efficient Sales Pitch with the Web of Things. In: ICEBE, 2013, pp. 377–384 (2013)
6. Mathew, S.S., Atif, Y., Sheng, Q.Z., Maamar, Z.: Building sustainable parking lots with the Web of Things. In: Personal and Ubiquitous Computing, 2013, pp. 1–13. Springer, Heidelberg (2013)
7. Mathew, S.S., Atif, Y., Sheng, Q.Z., Maamar, Z.: Ambient things on the Web. Journal of Ubiquitous Systems and Pervasive Networks (JUSPN) 1(1), 1–8 (2010, 2013)
8. Mathew, S.S.: Classifying and Clustering the Web of Things, University of Adelaide, School of Computer Science (2013), <http://hdl.handle.net/2440/83366>
9. Mathew, S.S., Atif, Y., Sheng, Q.Z., Maamar, Z.: The Web of Things - Challenges and Enabling Technologies. In: Bessis, N., Xhafa, F., Varvarigou, D., Hill, R., Li, M. (eds.) Internet of Things & Inter-cooperative Comput. Technol. SCI, vol. 460, pp. 1–24. Springer, Heidelberg (2013)
10. Mathew, S.S., Atif, Y., Sheng, Q.Z., Maamar, Z.: Web of Things: Description, Discovery and Integration. In: International Conference on Internet of Things and Cyber, Physical and Social Computing (iThings/CPSCoM), pp. 9–15. IEEE (2013)
11. Müldner, T., Miziolek, J.K., Leighton, G.: Succinct Access Control Policies for Published XML Datasets. In: ICEIS, vol. (1), pp. 380–385 (2008)
12. Osborn, S., Sandhu, R., Munawar, Q.: Configuring role-based access control to enforce mandatory and discretionary access control policies. ACM Transactions on Information and System Security (TISSEC) 3, 85–106 (2000)
13. Oh, S., Park, S.: Task-role-based access control model, Information Systems, vol. 28, pp. 533–562. Elsevier (2003)
14. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer Society 29(2), 38–47 (1996)
15. Security frameworks for open systems: Access control framework, Technical Report ISO/IEC 10181-3, ISO (1996), [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=18199](http://www.iso.org/iso/catalogue_detail.htm?csnumber=18199)
16. Thomas, R.K., Sandhu, R.S.: Task-based authorization controls (TBAC): A family of models for active and enterprise-oriented authorization management. In: DBSec, 1997, vol. 113, pp. 166–181 (1997)

Codes, Cryptology, and Information Security  
First International Conference, C2SI 2015, Rabat,  
Morocco, May 26-28, 2015, Proceedings - In Honor of  
Thierry Berger  
El Hajji, S.; Nitaj, A.; Carlet, C.; Soudi, E.M. (Eds.)  
2015, XXVI, 375 p. 58 illus., Softcover  
ISBN: 978-3-319-18680-1