Context-Sensitive Policy Based Security in Internet of Things

Prajit Kumar Das*, Sandeep Narayanan*, Nitin Kumar Sharma†
Anupam Joshi*, Karuna Joshi*, Tim Finin*
*University of Maryland, Baltimore County, Baltimore, MD, USA
{prajit1, sand7, joshi, kjoshi1, finin}@umbc.edu
†Indian Institute of Technology, Delhi, India
mcs142867@cse.iitd.ac.in

***Abstract*—According to recent media reports, there has been a surge in the number of devices that are being connected to the Internet. The Internet of Things (IoT), also referred to as Cyber-Physical Systems, is a collection of physical entities with computational and communication capabilities. The storage and computing power of these devices is often limited and their designs currently focus on ensuring functionality and largely ignore other requirements, including security and privacy concerns. We present the design of a framework that allows IoT devices to capture, represent, reason with, and enforce information sharing policies. We use Semantic Web technologies to represent the policies, the information to be shared or protected, and the IoT device context. We discuss use-cases where our design will help in creating an "intelligent" IoT device and ensuring data security and privacy using context-sensitive information sharing policies.**

I. Introduction

Since the advent of the Web, people have dreamt about accessing their toasters [1] and other electronic devices or things over the Internet. In recent years, there has been an exponential growth in the number of smart devices that are connected to the Internet. Gartner [2] predicts that by 2020 size of Internet of Things (IoT) will reach 21 billion. IoT systems, a network of devices that have the ability to sense, compute, communicate, and actuate, are also sometimes referred to as Cyber-Physical Systems (CPS). Owing to their ability to sense, compute, communicate and act, CPSs are enabling novel smart systems and applications in multiple domains.

Typically IoT devices consist of a computational unit with limited capability to meet cost and size restrictions. For instance a wearable fitness band has a typical CPU speed of 337 MHz. As a result, most IoT devices focus on ensuring the functional requirements and are typically not designed to focus on security. This makes them vulnerable, and we have seen multiple cyberattacks in the IoT domain in recent years. One of the most famous cyberattack incident, Stuxnet, targeted critical infrastructures in Nuclear power plants. It targeted a very specific Siemens SCADA (Supervisory Control And Data Acquisition) system [3] and reportedly ruined a significant number nuclear centrifuges, severely delaying the nuclear program of a country. Another CPS attack, the Jeep hack [4], involved a car driving at 70 mph in St. Louis which was remotely hacked into and its engine shut down. In 2014 Hacker News [5] published an article about one of the first proven cyberattack by Thingbots. In this attack, 750,000 malicious email communications were sent from 100,000 everyday consumer gadgets such as smart TVs, refrigerator etc. IoT devices thus open up new attack surfaces that puts users at risk in ways more complex than mere information loss or identity theft. Our effort is to create declarative, context senstive policies that allow these cyber-physical systems to operate securely.

The 2015 Gartner report [6] puts Internet of Things at the "peak of inflated expectations". This would mean that corporations would be paying more attention to developing and perhaps even deploying IoT devices on their corporate networks. Smart environmental sensors in buildings are being increasingly adopted and are now quite common in USA. In light of the IoT attacks, companies could soon be looking at severe security vulnerabilities through thingbots inside their corporate networks.

To mitigate these risks, the Internet of Things and applications deployed on them need to be "intelligent", and function in open and dynamic environments. They require a greater degree of decision making and autonomy. In the long-term, we envision societies of intelligent, adaptive, autonomous agent based IoT systems that exchange information about services offered and sought, and negotiate for information sharing constrained by policies they operate under. Such systems have to guarantee safety of users using them and be secure against unauthorized access, behavior modification and export of operational information, including sensor data over a network. This includes privacy of users personal data, like home temperature sensor data which can be utilized to ascertain home occupancy patterns.

Context-sensitive policies have long been used in smart environments for managing access control to system resources and data [7], [8]. Enterprises are no exception to policy based security systems. Depending on the context of usage, access rights may change. For instance, it might be permissible to send some data over the corporate VPN, but not if it is limited by information sharing policies of the IoT device. It might be fine to share building operational data with a cleaning robot that might need environmental information, but

978-1-5090-0898-816$31.00

this information should not be shared with an app on a random mobile device. Sharing camera feed with a security personnels authorized tablet computer might be fine in general, but not when inside a Sensitive Compartmented Information Facility (SCIF).

In this paper, we present a framework for securing IoT devices using declarative policies that are context-sensitive. We present a methodology for representing, capturing and inferring such fine-grained, context-sensitive information sharing policies for access control by using Semantic Web technologies. We represent policies using the Web Ontology Language (OWL) [9] and depict context using an ontology that allows us to define hierarchical contextual situations. This allows us to refine our policies depending on the requirements of a system. Our policies follow an attribute based access control (ABAC) model which uses location, activity, time etc. context attributes and roles as user attributes. Once the policies are specified, IoT devices are capable of reasoning over the information sharing policies and their contextual situation to draw inferences about access requests made by other IoT devices. In later sections we have shown that these policies can capture both Role based access control (RBAC) and ABAC based models of security. For the sake of simplicity we use the term CPS to mean both IoT and CPS throughout this paper. Rest of the paper is organized as follows. We discuss the related work from literature in Section II. Section III describes an overview of our system design. Section IV describes a few use-case scenarios and how they can be handled in our system, followed by our conclusions and a discussion of future research goals.

II. Related Work

Attacks on Cyber-Physical Systems are not new. In January 2000, Maroochy Shire Council's sewage control system [10] was attacked in Australia. This attack resulted in an anomalous plant behavior leading to flooding of the grounds of a hotel, a park, and a river with a million liters of sewage. In 2008 a Poland teenager [11] took control of a remote control and switched tram tracks resulting in multiple injuries. Medical equipments like implantable cardioverter-defibrillator (ICD) or automated implantable cardioverter defibrillator (AICD) [12], which can impact patient health and safety have been shown to be vulnerable. The literature is full of works that have identified IoT specific threat models [13], [14], [15] and have done analysis of security challenges in this domain. Roman et.al [13] have identified the security challenges and needs of the IoT domain. Some of these challenges include Authentication, Access Control, Network Security, Privacy, Trust and Fault Tolerance. There are several surveys that discuss interesting approaches for achieving security in CPSs.

In our previous work [16], we have discussed a number of possible attacks on a car CPS and have presented a data analytics based solution for it. In the current work we are focusing at a deeper level of security. By using context-sensitive policies we are controlling IoT system behavior at a granularity that has not been observed in such systems before.

There have been many efforts to model access control policies. XACML [17] which is based on XML specification language, is a general purpose authorization policy model. It enforces access control based on attributes. Rei [18] policy language is another effort which is based on deontic concepts. In Rei, credentials and entity properties like user, agent, etc are associated with access privileges. Another related piece of work is the Rein framework which builds on REI and also based on N3 rules. CWM reasoning engine is being used for distributed reasoning in Rein. KAoS [19] and ROWLBAC [20] are other works in the related area which are based on OWL. In this paper we use Attribute Based Access Control models and combine them with authorization policies using OWL. We decide what actions are permitted using inference based reasoning process.

III. System Overview

We have developed a policy based system for managing the security and privacy of IoT devices. The architecture design consists of two main system components, two Knowledge-bases (KBs) and two subsystems as illustrated in Figure 1. The first subsystem houses all device's sensors to detect environmental conditions, contextual situations etc. The second subsystem houses device's communication systems allowing it to connect with other IoT devices, remote monitoring, control units etc. The Sensor KB is used to store sensed data in the form of triples and Policy KB is used to store policies applicable to the system. The "world interaction subsystem" acts as an interface to the world and processes access requests for sensor data or system resource. The "Policy Decision" component is capable of reasoning over system attributes like sensed context, user roles and policy specification to draw inferences about access control requests. The "Policy control UI" component allows a system administrator to modify system policies. We discuss more about policy modification in Section III-C.

A. Cyber-Physical Systems

According to the National Institute of Standards and Technology [21], Cyber-Physical Systems or "smart" systems are co-engineered interacting networks of physical and computational components. As a pre-cursor to it, different Mecatronic and embedded systems existed which includes expert systems, automotive subsystems like Anti-lock braking systems etc. But the introduction of communication capabilities saw these different systems work coherently towards achieving a common goal and resulted in "smart" systems. Most cyber systems have four basic components. First basic component is the category of Sensors which monitor or sense different physical attributes of the environment. Next component includes the category Control systems which analyze sensor values, run intelligent algorithms on it and determine different actions to be performed. Next category include Actuators which are mechanical devices which can induce different actions on to the environment. Actuators are generally controlled by

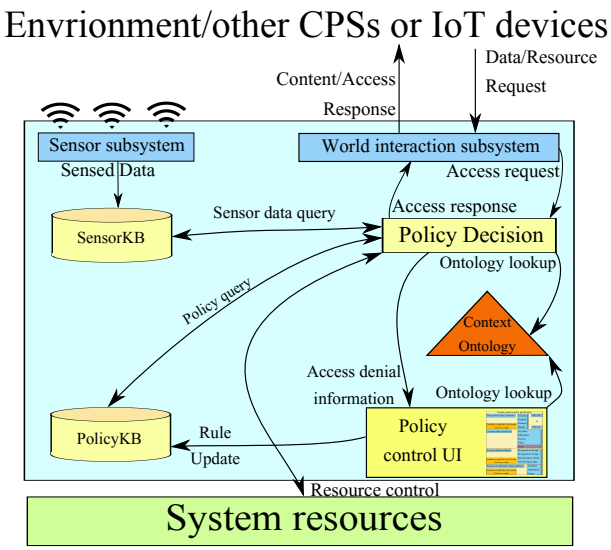

Fig. 1. System architecture for a Context-Sensitive Policy Based IoT security system

the control systems. The final category includes the communication channel and devices which enables communication between different components and interaction between other cyber-physical systems. Since the smart systems involves data exchange between different entities, both known and unknown sources, questions regarding what need to be shared and which entities have authorizations need to be properly represented, analyzed and enforced to maintain security and privacy.

B. Policy Modeling

Cyber-Physical Systems operate in a dynamic, distributed environment where their behavior changes dynamically depending on the context. The access control for such CPSs is governed by high-level and highly complex policies. The access control models describes entities needed to achieve restriction on information flow across the system. Examples of access control models include Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC) etc. These models when combined with formal policy specification language give the ability to write policies that describe entities and relationships in the system, how they affect access control, and how they are grounded out in models that are well understood in the security community.

1) Attribute Based Access Control Model: The Attribute-Based Access Control (ABAC) model provides access control based on the value and relationship among *attributes*. The ABAC model is a general framework which allows for more flexibility for policy specifications as any number of *attributes* can be added within the same extensible framework.

We have represented ABAC_α [22] and ABAC_β [23] models proposed by Jin *et. al*. While ABAC_α is a basic ABAC model which provides a unified framework to cover DAC, MAC, Flat RBAC and Hierarchical RBAC, the ABAC_β further improves the ABAC_α model by including contextual attributes. These models combine the benefits of DAC, MAC and RBAC and goes beyond their limitations. The models are based on *attributes* which are associated with users, subjects and objects. These *attributes* are used to capture identities and access control lists for DAC, security labels, clearances and classifications for MAC and roles for RBAC. One of the reasons to select ABAC model is that it solves the shortcomings of the core RBAC model as appropriate attributes such as `location, business hours` *etc.* can be added within a unified framework.

After specifying the authorization policy, dynamic computation of the authorization can be done at the time when the access request is made. As an example, the need to preassign the *roles* to users in RBAC can be avoided. Based on the authorization policy, the relationship is determined at run time and appropriate access decision is taken. These features makes ABAC suitable for automation required in the access control process for CPSs.

2) Representation of ABAC Policies in OWL: The Web Ontology Language (OWL) [24] provides an efficient way to represent policies formally. Although the language was primarily designed to represent the knowledge content of web information, it has been used to represent security policies [18] [25]. OWL provides an efficient and standard way to write complex ontologies. Entities associated with Cyber-Physical Systems and their relationships are quite complex in nature and OWL helps to capture them easily.

OWL representation of ABAC_α policies have been presented in [25]. In this work, basic constructs (`User`, `Subject`, `Object`, `Permission`) are defined as OWL classes. OWL properties are used to define `User Attribute`, `Subject Attribute` and `Object Attribute`. For example, OWL can be used to define location facilities available in a CPS like `Smart Car` identified by tag *car1234* as:

```
car1234 a abrbac:Object;
  hasLocationFacility GPS, Navigation.
```

Access to a resource is granted if there exists a consistent relationship among user, subject and object attributes for a particular permission. The enforcement of authorization rules happens dynamically at run-time by an inference based reasoner. For that purpose rule-based OWL reasoners are available for embedded devices from Seitz et. al. [26]. [25] shows that ABAC model can be used to enforce DAC, MAC, Flat RBAC and Hierarchical RBAC based policies. For example, the FLAT RBAC to ensure that *the location of an object (car) can only be accessed by those subjects whose roles are listed in the allowed object roles list* is represented in OWL as:

```
{ ?A a abrbac:RequestedAction;
          abrbac:subject ?S;
          abrbac:object ?O;
          abrbac:permission ?P.
  ?P rdfs:label "locationAccess"^^xsd:String.
  ?S abrbac:srole ?r.
```

```
  ?O abrbac:orole ?r.

} => { ?A a abrbac:PermittedAction }.
```

We further build from here to incorporate contextual attributes. In general, context attributes cannot be associated with user, subject or object. Therefore we define context separately as a basic OWL class:

```
Context a owl:Class.
```

Any request to access some resource may have a particular context. These may include the day, date and time at which the request is made. Other examples are *an activity in progress*, *presence of other users (or subjects)*, *location*, *place etc*. The *Platys* [27] ontology comes handy for making the context more formal and structured.

The *Platys* [27] ontology associates a user with a device. The device has a position which is mapped to a geographic place and a conceptual place. As an exmaple, `Mrs. Smith` (*user*) drives her `smart car` (*device*) to `NASA's Goddard Space Flight Center` (*geographical place*) which happens to be her `workplace` (*conceptual place*). The ontology also defines the concept of activities which is crucial to mapping positions to places. This approach helps combining activities, their occurrence time and place, involved participants and users, devices *etc*.

The *Platys* activity is associated with the context as an `object property`:

```
contextActivity  a owl:ObjectProperty;
  rdfs:subPropertyOf CA;
  rdfs:domain Context;
  rdfs:range Platys:Activity.
```

The *Platys* activity has associated participants which, in-turn, has associated users. By incorporating *Platys* in our representation, we are now able to write policy rules like: *parking to a location is always granted (irrespective of the privilege of the owner) if Mrs. Smith is present in the car*:

```
# Authorization Policy for parking in presence
of Mrs. Smith
{ ?A  a abrbac:RequestedAction;
      abrbac:subject ?S;
      abrbac:object ?O;
      abrbac:permission ?P;
      abrbac:context ?C.
  ?P rdfs:label "parkingAccess"^^xsd:String.
  ?C abrbac:contextActivity ?cAct.
  ?cAct Platys:has_participant ?p.
  ?p Platys:has_user data:MrsSmith.
} => { ?A a abrbac:PermittedAction  }.
```

C. Policy Control

The Policy Control unit is meant for use by a privileged user (administrator). For CPSs it is important not only to be able to decide who gets access to what in what context but also to capture any change in behavioral need. Take for example, we have a high level policy specifying that "at `home` access to car should be given to `all family` members", essentially people who have a key. Such a policy leaves out a loophole that allows a child to use the car unsupervised. However, the owner may be using a car that has cameras and other biometric sensors. In short, a car of the future should be intelligent enough not provide access to unauthorized persons. So a default policy that gives access to anyone who has the car keys would require modifications. A simple UI for such a context-sensitive policy specification can be seen in Figure 2. An example of a modified policy could be "at `home` access to car should be given to `adult family` members". As we can see the UI we have presented would allow a user to make a policy change easily. We use Platys ontology to formally define context, which is then used to feed the options of our user interface.

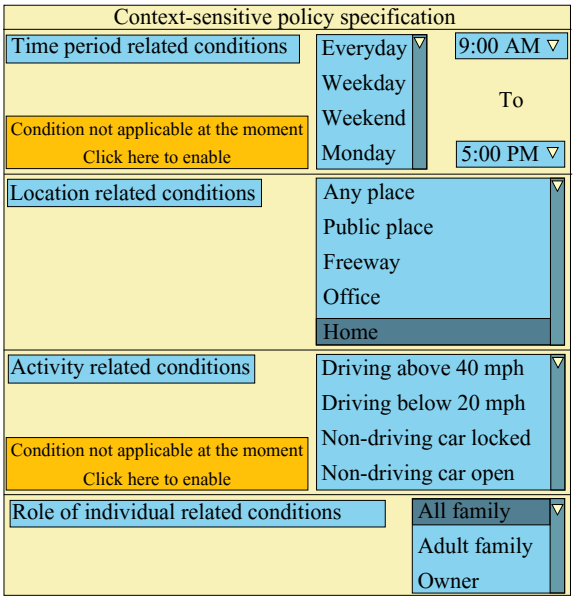

Fig. 2. Generalize or specialize context for a policy

Current cars come with displays that transfers a mobile's interface to the car's using Android Auto or Apple Carplay. We take advantage of such interfaces to handle user input. Modified policies can then be transferred to the CPS using the Proximity Beacon API from the Google Beacon platform [28], which allows us to broadcast a Beacon describing the capabilities and constraints of each device as a knowledge graph. We may either broadcast an Eddystone-URL, the backbone of the Physical Web, or directly attach the implication as an RDF payload.

Through our ontology we have defined the notion of a hierarchical context model. As we are using an ABAC model for our policies, we use a context ontology and context as an attribute for the policies. Thus using the hierarchical context model we are enabling a user to generalize or specialize context attributes for a policy. We can define highly complex policies using combinations of multiple context pieces. We enable users to do this by allowing them to add or remove context attribute constraints for a policy. Our system can therefore reason over and infer access control outcomes for

complex and fine-grained context-sensitive situations.

As can be seen in Figure 1, in our system, any access denial information is passed on to the policy control unit. Every denial instance may allow us to detect a plausible attempt at breaching the system but it can also indicate a legitimate request. Therefore, we use denial results to query administrators and further improve the policy specification or security of the system. Once a policy has been confirmed repeated denial of requests could potentially indicate an attack. Once policy modifications have been submitted by an admin user, we update the PolicyKB with new rules and the system resets all denial counts and starts the process of monitoring all over again.

IV. Use Case Scenarios

Before we explain our use-case scenario we need to specify that some of the things we have mentioned here already exist in real life, and some have been "invented", so to speak, to demonstrate the extent of "intelligent" operations. As is the case in our system, we use an attribute based access control paradigm in our policy definition and we have explained how our proposed architecture is incorporated into the CPS. The car is no longer just a combination of mechanical devices and a 'bit" of electronics rather it is a smart cyber-physical system capable of executing context-sensitive information sharing policies and protecting the security and privacy of the car and it's user. A car CPS is thus an aggregation of dozens of separate units, each unit consisting of a number of sensors, actuators and a control unit. Examples of some of these units include, safety systems like Anti-lock braking systems (ABS), Automatic Lane Assist, Adaptive Cruise Control. They also include in-car GPS navigation systems & entertainment systems like audio player systems, video player systems, satellite radio systems or even Electronic Toll Collection (ETC) systems. With the advent of high speed cellular and wireless networks, cars have been provided with external connectivity thus making vehicles capable of interacting with systems that are external to the car. Such external systems may include a simple toll collection sensor or another CPS with a plethora available services. Hence, in presence of such external systems, we discuss a scenario where a vehicular CPS becomes a source of data for them. Naturally, we have to take into consideration a scenario where security and privacy of such data might be affected. Specifically if such data may include Personally Identifiable Information (PII) about the vehicle and it's user.

In our use case scenario, we are going to share a vision of a "smart" car in the near future and driving it around in a "smart" environment. In this scenario, Mrs. Smith (our fictitious user) drives her car to her office in the morning. The office has a "smart" parking lot that is capable of detecting the presence of a car and requests for employee authentication. From the car's perspective sharing employee identity is dependent on temporal context like time and day of week and location context like at "office" parking lot. Upon receiving the request for sharing identity information, the car's reasoning unit can reason over the current context information and the current information sharing policy to authorize the identity sharing operation with the "smart" parking lot's system. Now the parking lot's system takes the employee identity and upon completion of the authentication process feeds back the GPS coordinate of an open parking slot back to the car. In order to do that it may also use some contextual data and information sharing policy to reason over and authorize the data sharing operation.

In case Mrs. Smith is visiting an office where she is not employed, a similar negotiation might be possible. In this case a "smart" parking lot system would simply request payment information or parking validation information from the car. For such a scenario Mrs. Smith has a different policy that allows payment approval based on the location being identified as a parking location. Using location context the reasoner will be able to identify the current location as a parking lot by looking up a connected KB for parking locations. As a result, Mrs. Smith's credit car information would be shared with the parking lot's system. If the location is not determined to be a parking lot the stored credit card information would not be shared.

Now in the evening, Mrs. Smith drives back home in her car. Since she can be identified by the car as the owner by referring to it's internal KB, it will present private information like car health status, maintenance needs etc. However, if the car detects someone else driving, (Mr. Smith perhaps) it will only display critical information like gas level and allow access to the GPS but not to the GPS history. If the system detects children in the car, it can automatically activate child lock feature for it and implement parental control features on the car's entertainment options by using alternate audio playlists, video streams etc.

Nowadays, cars have self-parking features [29]. In our use-case scenario, upon reaching home, Mrs. Smith leaves the vehicle in the driveway. The car then communicates with a "smart" garage system installed in her home. After the standard authentication process, described earlier, the garage door opens and the car is able to detect if the garage is safe to enter (no kids around!) and park itself inside. If there is a electric charging post and the car is an electric car, the charging system requests access to the car's charging port. By reasoning over location context, the car is capable of providing the access to the port because it determines itself to be at "home".

Our final use case scenario deals with the policy modification point of view. Imagine that the default policy installed on Mrs. Smith's car by the manufacturer stated that only the driver should have access to the operational systems during driving if the car location was "freeway" and car's was "driving above 40 mph" speed. Mrs. Smith decides that the kids need to watch their cartoons while on a drive so she modifies the policy to create two different policies. The first one states "access to `driving operations` to be given to `driver` when on the `freeway` and `driving above 40 mph`". Second rule states that "access to `entertainment system` to be given to `all family` when on the `freeway` and `driving`

above 40 mph.

V. Conclusions and Future Work

The IoT field is exploding with novel smart systems and applications in multiple domains. Most of these systems and applications leverage the computation and communication capability of such systems that allows information sharing and collaboration. Information leakage from CPSs or behavior modification of such systems can have dangerous real life impacts. Upon doing a survey of the literature we found a number of attacks have already been mounted on CPSs and how they affected human lives. Hence in this paper, we have proposed the design of a system that allows context-sensitive policy based security to control and protect information sharing operations among CPSs. Our system design creates a middle-ware that is capable of executing such policies and thus protect security and privacy of user and her data. We use Semantic Web technologies to represent our policies and to reason over contextual attributes and user role attributes to determine outcomes of access control requests. We use a context ontology to allow easy policy refinement. Due to dynamic and open environments that IoT systems are deployed in, their access control policies maybe highly complex and we are able to capture that by using Attribute Based Access Control (ABAC) model represented in OWL. We also describe a few use case scenarios that shows how access control decisions can be made in such a system.

As part of future work, we would like to evaluate performance of CPS systems when a reasoning system executes access control policies. Detecting suspicious events at run-time could be another interesting area of research. The introduction of Proximity Beacon API from Google, have made sharing of information like policies, capabilities, services etc. easier for CPSs. However, self-organization and interoperability between a diverse group of CPSs is still a challenging goal.

Acknowledgment

Support for this work was provided by NSF grants 0910838 and 1228198, a Supplement from DoD to NSF award 1439663, and funds from the Oros Family Professorship.

References

[1] V. G. Cerf, "Prospects for the internet of things," *XRDS*, vol. 22, no. 2, pp. 28–31, Dec. 2015. [Online]. Available: http://doi.acm.org/10.1145/2845145

[2] N. Eddy, "Stress-free parking," November 2015. [Online]. Available: http://www.informationweek.com/mobile/mobile-devices/gartner-21-billion-iot-devices-to-invade-by-2020/d/d-id/1323081

[3] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, Nov 2011, pp. 4490–4494.

[4] K. Kochetkova, "Shock at the wheel: your jeep can be hacked while driving down the road," July 2015. [Online]. Available: https://blog.kaspersky.com/remote-car-hack/9395/

[5] S. Khandelwal, "100,000 refrigerators and other home appliances hacked to perform cyber attack," January 2014. [Online]. Available: http://thehackernews.com/2014/01/100000-refrigerators-and-other-home.html

[6] I. Gartner, "Gartner's 2015 hype cycle for emerging technologies identifies the computing innovations that organizations should monitor," August 2015. [Online]. Available: http://www.gartner.com/newsroom/id/3114217

[7] H. Chen, F. Perich, T. Finin, and A. Joshi, "Soupa: standard ontology for ubiquitous and pervasive applications," in *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, Aug 2004, pp. 258–267.

[8] L. Kagal, T. Finin, M. Paolucci, N. Srinivasan, K. Sycara, and G. Denker, "Authorization and privacy for semantic web services," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 50–56, Jul 2004.

[9] S. Bechhofer, "Owl: Web ontology language," in *Encyclopedia of Database Systems*. Springer, 2009, pp. 2008–2009.

[10] J. Slay and M. Miller, *Lessons learned from the maroochy water breach*. Springer, 2007.

[11] J. Leyden, "Polish teen derails tram after hacking train network," *The Register*, vol. 11, 2008.

[12] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 129–142.

[13] M. Abomhara and G. M. Koien, "Security and privacy in the internet of things: Current status and open issues," in *Privacy and Security in Mobile Systems (PRISMS), 2014 International Conference on*. IEEE, 2014, pp. 1–8.

[14] K. Zhao and L. Ge, "A survey on the internet of things security," in *Computational Intelligence and Security (CIS), 2013 9th International Conference on*. IEEE, 2013, pp. 663–667.

[15] 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.

[16] S. N. Narayanan, S. Mittal, and A. Joshi, "Using data analytics to detect anomalous states in vehicles," *arXiv preprint arXiv:1512.08048*, 2015.

[17] S. Godik, A. Anderson, B. Parducci, P. Humenn, and S. Vajjhala, "Oasis extensible access control 2 markup language (xacml) 3," Tech. rep., OASIS, Tech. Rep., 2002.

[18] K. Lalana, "Rei: A policy language for the me-centric project," *TechReport, HP Labs*, 2002.

[19] J. M. Bradshaw, A. Uszok, M. Breedy, L. Bunch, T. Eskridge, P. Feltovich, M. Johnson, J. Lott, and M. Vignati, "The kaos policy services framework," in *Proc. 8th Cyber Security and Information Intelligence Research Workshop*, 2013.

[20] T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough, and B. Thuraisingham, "R owl bac: representing role based access control in owl," in *Proceedings of the 13th ACM symposium on Access control models and technologies*. ACM, 2008, pp. 73–82.

[21] NIST, *NIST CPS*, 2016 (accessed February 1, 2016). [Online]. Available: http://www.nist.gov/cps/

[22] X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering dac, mac and rbac," in *Proceedings of 26th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy (DBSec 2012)*, Paris, France, July 2012.

[23] X. Jin, "Attribute-based access control models and implementation in cloud infrastructure as a service," Ph.D. dissertation, The University of Texas, San Antonio, May 2014.

[24] M. Dean and G. Schreiber, "Owl web ontology language guide," W3C Recommendation, http://www.w3.org/TR/owl-guide/, 2004.

[25] N. K. Sharma and A. Joshi, "Representing attribute based access control policies in owl," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, California, USA, Feb 2016, pp. 333–336.

[26] C. Seitz and R. Schönfelder, *The Semantic Web – ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. Rule-Based OWL Reasoning for Specific Embedded Devices, pp. 237–252. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25093-4_16

[27] L. Zavala, P. K. Murukannaiah, N. Poosamani, T. Finin, A. Joshi, I. Rhee, and M. P. Singh, "Platys: From position to place-oriented mobile computing," *AI Magazine*, vol. 36, no. 2, 2015.

[28] Google, "Mark up the world using beacons," March 2016. [Online]. Available: https://developers.google.com/beacons/

[29] BMW, "Gartner's 2015 hype cycle for emerging technologies identifies the computing innovations that organizations should monitor," 2013. [Online]. Available: http://www.bmw.com/com/en/insights/technology/connecteddrive/2013/driver_assistance/intelligent_parking.html