A Smart-Farming Ontology for Attribute Based Access Control

Sai Sree Laya Chukkapalli*, Aritran Piplai*, Sudip Mittal†, Maanak Gupta‡, Anupam Joshi*
* *University of Maryland Baltimore County*, Baltimore, MD, USA
{saisree1,apiplai1,joshi}@umbc.edu
† *University of North Carolina Wilmington*, Wilmington, NC, USA
mittals@uncw.edu
‡ *Tennessee Technological University*, Cookeville, TN, USA
mgupta@tntech.edu

***Abstract*—With the advent of smart farming, individual farmers have started adopting the concepts of agriculture 4.0. Modern smart farms leverage technologies like big data, Cyber Physical Systems (CPS), Artificial Intelligence (AI), blockchain, etc. The use of these technologies has left these smart farms susceptible to cyber-attacks. In order to help secure the smart farm ecosystem in this paper, we develop a smart farming ontology. Our ontology helps represent various physical entities like sensors, workers on the farm, and their interactions with each other. Using the expressive ontology we implement an Attribute Based Access Control (ABAC) system to dynamically evaluate access control requests. Furthermore, we discuss various use cases to showcase our access control model in various scenarios on a smart farm.**

***Index Terms*—Smart Farming, Ontology, Cybersecurity, Access Control**

I. Introduction

With the world population projected to grow to 9 billion by the year 2050, considerable investments are being made in the field of smart farming [1], [2]. With the advent of smart farming, individual farmers are ‘reaping’ the benefits of integrating precision agriculture technology to better manage their farming operations and improve productivity. Farmers have started adopting the concept of *Agriculture 4.0* and precision agriculture, which leverages technologies like big data, cyber physical systems (CPS), artificial intelligence, blockchain, etc. [3], [4]. The development of smart farms have given individual farmers a method to monitor and manage their farms effectively.

In comparison to existing farming practices, smart farming involves communication between deployed on-field smart sensors and devices which work together to provide an efficient farming experience. These developments of using CPS and data assisted technologies have improved the overall crop yield [4].

However, the use of such connected and internet enabled technologies in the smart farming ecosystem has exposed it to potential cyber-attacks and vulnerabilities. With the increase in the number of attack surfaces, significant hurdles exists that pose a threat to the agriculture sector [3]. These attacks can exploit and remotely control on-field sensors, autonomous tractors, or aerial vehicles. Such developments call for specific security solutions to protect the smart farming ecosystem.

The problem has been further exacerbated by the development of various big data and artificial intelligence applications specifically designed for the smart farming use-case. Nowadays, farmers can use decision support systems to know the best times to plant certain crops, given farm specific factors [5]. These applications can help farmers understand the quality of their crops, enable them to hire workers and buy the right equipment for their farms. Specific technologies need to be developed to secure these big data and artificial intelligence systems.

In this paper, we create a smart farming ontology and use it to develop an Attribute Based Access Control (ABAC) [6] system. We begin by creating a smart farm ecosystem to encode farm specific sensors and interactions. Our system architecture consists of physical entities that include on farm sensors like automated sprinklers, soil moisture sensors, temperature sensors, etc. Machinery like autonomous tractors, reapers, harvesters, trucks; farm labor and workers all connected to the internet and the cloud by a gateway hub set up by the farm owner. We also explain in detail various interactions that happen between the owner, workers, sensors and vehicles that are present on the farm. Using this architecture and an interaction model, we created a smart farm ontology and implement Attribute Based Access Control (ABAC) for the farm. We also discussed various use case scenarios and how access control decisions are made using our smart farming ontology.

The structure of the remaining paper is as follows- Section II discusses some related work on smart farms, various attacks and some methods to secure a smart farm. Section III presents our system overview, its architecture, various interactions and our smart farming ontology. Section IV presents how our system handles various security use case scenarios. Finally, Section V concludes the work.

II. Related Work

Rapid technological advancements in the domain of Internet of Things (IoT) has paved a path for smart farming. A smart farm can be considered as a framework to manage interactions for performing various farming related tasks based on context obtained from the data collected in real time [7]. In this section we describe some relevant work on the smart farming ecosystem, attacks on smart farms and access control.

A. Smart Farming Ecosystem

It is an evolving cyber physical domain and is getting wide acceptance in the agriculture dependent communities. Also referred to as *precision agriculture*, the concept involves smart sensors spread across farmlands for improving agriculture practice with minimal human and natural resources. These sensors provide data driven applications which enable farmer to make optimal decisions for the farms. As discussed by Fountas et al [8], incorporation of smart sensors can help to reduce the damage done to the crop. Different types of sensors that can be used in farms include global positioning system (GPS) sensors which collect the latitude, longitude, altitude and environmental data of the farm for crop mapping [9]. The humidity and temperature sensor have helped in identifying the germination issues or risk of over irrigation based on the values collected causing negative impact as discussed by Cancar et al. [10]. Data collected from these smart sensors has lead to smart production by reducing excess usage of fertilizers, water in the farm as presented by Sabiha and Rahman [11]. A cloud based framework was proposed by Yang et al. [12] where large data collected from the sensors in the farm can be stored in the cloud for faster computation which helps in uniformed decisions.

B. Attacks on Smart Farms

As the sensors deployed in the farm get connected to the internet and communicate with each other, they are exposed to security threats similar to other IoT domains. For example, in the year 2015 cyber attack on the Ukrainian power grid had a power outage due to False Data Injection (FDI) which lead to loss of service for 225,000 consumers [13]. In 2011, a hacking attack named 'The Night Dragon' [14] leaked large amount of critical data regarding gas and oil from the petrochemical companies including Shell, BP, Exxon Mobile. Therefore, increase in number of sensors that communicate will increase the need for a more secure environment else risks like Man in the Middle attack become more likely to occur. If such attacks happen in the smart farm it will lead to a huge loss and damage to the crop which is irrecoverable for a farmer. A report by the U.S. Department for Homeland Security [3] describe cybersecurity threats and vulnerabilities in smart farming. Jahn et al. [15] have also discussed cybersecurity implications when using sensors in the agriculture sector. Lopez et al [16] have also elaborated similar security issues which occur for the IoT sensors deployed in smart farm .

C. Access Control Solutions

Access control mechanisms help in controlling the operations that can be performed by entities on different objects in the system. Similar access control approaches work for IoT systems as well. Discretionary Access Control (DAC) [17] is used where the owner of the objects can either grant access or revoke access. On the other side Mandatory Access Control (MAC) [18] provides access based on the classification and clearance of subjects and objects in the system. Role Based Access Control (RBAC) [19] requires users to be assigned to different roles to get the associated permissions. However, the problems of role explosion limits its use to enterprise systems only. Further, how the notion of role fits into distributed IoT domain where sensors have different administrative is challenging. Attribute Based Access Control (ABAC) [6], [20] is gaining more popularity, and has been used to overcome the issues faced in DAC and RBAC. Further, the fine grained context aware policies makes it more flexible and an optimal solution for cyber physical domains like smart farming. The Web Ontology Language (OWL) [21] helps in writing complex ontology and reasoning the entities relations and is being used to represent security policies. Rathod et al. [22] created an ontology based system for a popular cloud orchestration platform. Joshi et al. [23]–[25] have also created ontology based ABAC system specifically designed for cloud storage services. In our paper, we discuss and develop an OWL [21] based ABAC security policies for smart farming sensor communication.

III. System Overview

We developed a smart farming ecosystem which is a secure environment for the farm owners to monitor their farms. The proposed system offers immense value to the individual farmers as it allows them to take decisions based on the information collected and stored in the cloud infrastructure from sensor devices spread across the farm. Data in the cloud contain details regarding the availability of agriculture machinery like tractors, reapers, etc. and also availability of farm labors on a daily basis for a particular geographical area. Our proposed model can be integrated in an individual farm by the owner to manage several operational functions in a secure way to increase the crop yield, allocate workers to perform various tasks during the peak season, etc.

In this section we describe our smart farm ecosystem as shown in Figure 1. We have divided this section into three modules. The first module elaborates the architecture of system that has three layers. The first layer contains physical entities, second layer discusses digital twins and third layer is the cloud layer which has a representation graph. The second module is interactions, elaborating all the possible communications that occur between different types of sensors present in the farm. In the third module, we discuss our smart farming ontology. The following part describes each of the modules.

A. Architecture

1) Physical Entity: The physical entities of our smart farming ecosystem, as shown in Figure 1, have been divided into various categories of IoT sensors. There are static sensors which are used in the farm such as automated sprinklers, soil moisture sensor, temperature sensor, which are represented as

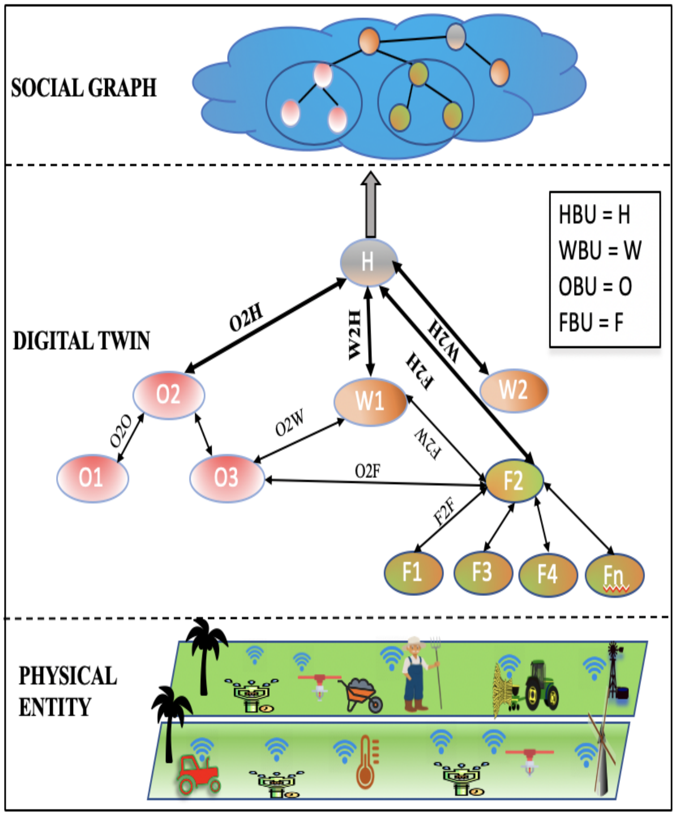

Fig. 1. Smart Farm Ecosystem Architecture & Interactions.

Farm Based Unit (FBU). Machinery like autonomous tractors, reapers, harvesters, trucks, etc. which come under the category of movable equipment are represented as *On-Board Unit* (OBU). Farm labor and workers are provided with mobile devices and computer systems to operate farm sensors and for communication, are represented as *Worker Based Unit* (WBU). Farmer who interconnects all these sensors via internet and monitor the actions of sensors by setting up a gateway hub is represented as a *Home Base Unit* (HBU) which is connected to the cloud.

Once connected to the central cloud, data analytics and machine learning assisted data driven applications provide suggestions and services to the farmer. For example, values from temperature, soil moisture sensors stored in cloud can be used along with weather forecast in the area (like heavy rainfall shower in next 24 hours) to help farmer to regulate the automated water sprinkler. In this way excess water in the farm could be avoided which could damage the crop yield.

2) Digital Twin: The digital twins are virtual replicas of physical entities. They play a very important role in enhancing the security of real-time physical sensors by monitoring and optimizing. Digital twins monitor the data that is authorized to be shared and to what extent in order to maintain security and privacy. Interaction between different sensors are represented as edges, this along with corresponding timestamp logs are stored in the cloud. These can accessed by the owner with the help of HBU. For example, if temporary labor is hired to work in the farm, the digital twin will not let him access the past data of the physical sensors to prevent data leakage. Optimization is also done by digital twins to use the resources in efficient way by training machine learning models from the data captured. For example, farmers can take better decisions regarding the number of workers required to work in the farm to save the cost of human resources and take measures to protect the crop from predicted crop diseases based on the past information stored.

3) Representation Graph: The top most layer in our architecture that include nodes and edges. This is a graph like structure (generally created in the cloud) that helps us represent and monitor interactions between various cyber physical systems on the smart farm. Here, each node refers to a physical entity and the edges are referred to as the connection between the nodes for information exchange. The communication that happens between the nodes is collected from the representation graph and stored in the cloud. Therefore, the cloud has collection of interactions between entities such as FBU-FBU, OBU-FBU, and WBU-FBU. Access to this content is given only to the farm owner for visualization and to monitor all the types sensors present in the farm.

Optimization of the representation graph is done in the cloud to eliminate redundancy and detect abnormal events. For example, whenever an interaction happens between the automated water sprinkler and temporary worker whose access to the farm has expired, an event automatically is stored in the cloud along with the timestamp which helps the farmer to identify anomalous actions performed by the worker in the farm and block those actions immediately from further damaging the crop. The data collected from the past can also help the farmer to plan his next tasks based on the analysis done in the cloud. The representation graph helps us create our ontology in Section III-C.

B. Interactions

A smart farm will have several types of interactions among various entities as follows-

FBU-FBU: Physical sensors and actuators in the farm are interconnected. Communication between the FBUs is bidirectional. FBU contains information about current status and past actions performed, stored in the representation graph. This helps the FBU to exchange information, interact with OBUs, WBUs and HBUs. If FBUs perform an action then, it is based on the instruction received from HBU or authorized WBU. The HBU in the farm gets updated whenever there is a status change in its local FBUs.

OBU-OBU: OBUs are movable equipment that can interact with the other physical OBUs when present in the same geographical region. Digital twins are created for every corresponding OBU. The OBUs store the interactions that happened between the physical OBUs or FBUs in the representation graph. WBUs and HBUs have access to the representation graph stored in OBUs, to enable them to control and monitor the OBUs. The representation graph stores every interaction of an OBU with the other OBUs and FBUs, which is saved at the HBU.

OBU-FBU: When OBUs want to perform actions such as crop harvestation, sowing seeds, dispersing fertilizer, etc. they interact with various FBUs to gather data. The representation graph contains all the interactions that occurred between the FBUs which can be accessed by the OBUs. The information exchanged between the OBUs and FBUs are also stored in representation graph. This plays an important role in identifying the nature of interactions that have occurred for performing an action.

FBU-WBU: FBUs present in farm are usually operated by WBUs to perform specific actions which can be based on weather or other factors at any time. A WBU can interact with FBUs only when granted security permission. A WBU requesting an operation on FBU is stored as an interaction in the representation graph to increase reliability of the system. FBUs status is sent to the WBU whenever there is any interaction to keep the workers updated.

OBU-WBU: Workers operate the OBUs in order to perform various functions in the farm as needed. The exchange of information between the OBUs and WBU create a virtual replica which is stored in the representation graph, that can be accessed by the owner of the farm. The graph keeps updating when an interaction happens and is stored in the HBU. WBUs can acquire only the present information related to the OBUs from the time they are given access. This way the WBUs will not be able to collect past data from the OBUs (see Section IV for an example).

FBU-HBU: The WBUs present in the farm can interact with the FBUs only when permission is granted by the HBU. The HBU has permanent access to the all the FBUs in the farm. HBU stores the representation graph and all the information exchanged with the FBUs. This helps the HBU to access the historical information of the FBUs to help them analyze and take decisions based on the different scenarios.

OBU-HBU: When HBU receives the OBU-OBU interaction it stores it in the representation graph to keep a track of the OBUs operating on the farm. OBUs and HBUs are connected through internet. OBUs operation permissions is only granted by HBU till the OBUs are present in the vicinity of the farm. If the OBUs are away from the farm then access to interact with other entities or to operate in the farm is denied.

WBU-HBU: HBU plays an important role in the network for building the smart farm ecosystem. The HBU can decide to give temporary or permanent access permissions for OBUs and FBUs, to WBUs. If the WBUs are given temporary permission, they can access only the data stored from the units for the time period specified, or based on their labor contract agreement. All the interactions between WBUs and HBU are stored in the cloud in the form of a representation graph.

C. Ontology

Using the system architecture and various interactions described in section III-A & III-B, we have created an ontology schema shown in Figure 2. It contains some of our major classes that are part of our smart farm ecosystem. In our ontology the *Owner* class is the subset of *Person* class and instances of this class configure various access policies over other farm specific classes like, *Home_Unit* (HBU), *Farm_Unit* (FBU), *OnBoard_Units* (OBU), *Worker* (WBU) present in the ecosystem. Another important class in our smart farm ecosystem is *AvailabilityofWorkers*. This class helps the system determine idle instances of the *Worker* class available for a particular time periods and it's readiness to engage in various interactions with other class instances on the farm. The owner of the farm can choose accordingly the worker to perform various duties in the farm.

The interaction between various sensors can be easily represented through various object and data properties. We also use the properties to determine the access control as discussed in Section IV. Some of the important properties are:

- *readAccess*(int:SmartFarm, X): This data property allows to read data from the sensors present in the farm only if they have boolean value X that states True.
- *hasAccessPermission*(int: SmartFarm, Y):This data property states that access control to particular devices in the SmartFarm can be authorized only if the boolean value of Y is True and the time context of the request is within the permitted time frame listed in "accessHour".
- *hasOperationPermission*(int: SmartFarm, Z): This data property makes sure that the boolean value of Z is True in order to give permission to operate particular devices in SmartFarm.

Next, we describe our Attribute Based Access Control system implemented using our smart farm ontology. We also discuss how we use the above mentioned data properties to create various SWRL [26] rules that enable us to dynamically compute access permissions.

IV. Access Control Use Cases

In this section, we describe an Attribute Based Access Control (ABAC) system built on our smart farming ontology. The goal of this system is to help farmers create and enforce access control rules for their smart farms. We discuss multiple access control scenarios that happen on a farm and various rules written in the Semantic Web Rule Language (SWRL) [26] that can be used to determine access. SWRL rules contain two parts, antecedent part (body), and a consequent (head). The body and head consist of conjunctions of a set of 'atoms'. Informally, a rule may be read as meaning that if the antecedent holds (is "true"), then the consequent must also hold. The knowledge graph and the ontology has been represented in the Web Ontology Langauge (OWL) [27]. We have also considered scenarios where there is a threat to data privacy and security. Our security policies and rules conform to the Attribute Based Access Control (ABAC) model, where access decisions are dynamically computed using an entity's fine grained attributes.

To aid the access control module we include in our implementation a '*context*' class. The context class has been used to include contextual attributes such as time, day of the week,

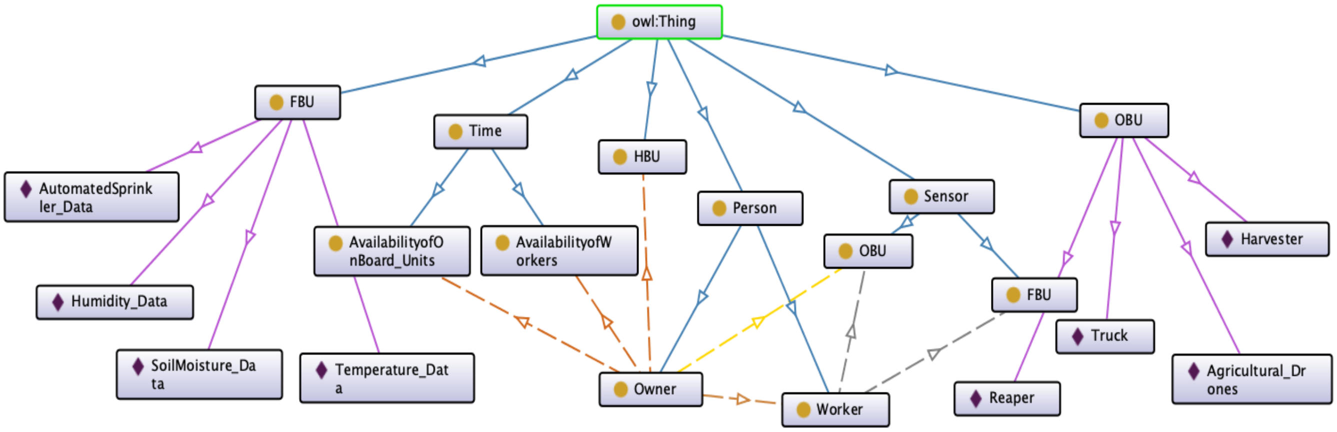

Fig. 2. Some of the relevant classes and properties in our smart farming ontology.

activity, etc. The inclusion of the context class enables the system to process access requests that need these types of information.

In order to obtain various device parameters like location, IMEI numbers, power consumption, etc. for various FBUs, OBUs, and WBUs, we use the ‘*Platys*’ ontology [28]. This ontology enables us to monitor and manage various devices deployed on the smart farm at any given time. Also, Platys enables us to efficiently represent physical locations on the smart farm, enabling us to add location based access control on the smart farm. We can include specific rules that allow access control based on the situational and location awareness.

For our smart farm ecosystem, we implement the following policy rules in our representation framework based on the ontology. These rules have been discussed in the form of various scenarios -

A. Scenario 1 - Data Reading Permissions

In this scenario a worker using a Worker Based Unit (WBU) like a mobile device with an IMEI number, tries to read the data collected from a Farm Based Unit (FBU), like a Humidity sensor. The access request is decided by the Home Base Unit (HBU) which has been configured by the farm owner. The SWRL rule for the above scenario is as follows -

```
# Access Permission for reading farm
sensor(Humidity_Data) data in presence of IMEI Number

{ ?A a abac:RequestedAction;
      abac:subject ?S;
      abac:object data:Humidity_Data;
      abac:permission ?P;
      abac:context ?C.
  ?P rdfs:label "readAccess"^^xsd:String.
  ?C abac:contextActivity ?cAct.?cAct
platys:has_participant ?p.
  ?p platys:has_user ?u.
  ?u platys:owns "IMEI345678234890345".

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

From the above policy rule the Worker Based Unit *S*, creates a request *A*, to access the data collected from an FBU. The request *A* will only be permitted if the WBU is allowed to access the FBU data.

B. Scenario 2 - Worker (WBU) & On Board Unit (OBU) Access Permissions

In this scenario, if the owner is leasing OBUs temporarily based on their availability for performing several functions in the farm. The rule `hasAccess` determines that the OBU can be operated by the owner and authorized workers only after proper authentication and access computation. Both the owner and the authorized worker automatically loose access after lease expiration. This way a previous user of the On-Board_Unit cannot exploit the farm by gaining access. For this case, we can write the rule as:

```
# Contol the activity on tractor in presence
of owner based on time and day

{ ?A a abac:RequestedAction;
      abac:subject ?S;
      abac:permission ?P;
      abac:context ?C.
  ?P rdfs:label "hasAccessPermission"^^xsd:String.
  ?C abac:contextActivity ?cAct.?cAct
platys:has_participant ?p.
  ?p platys:has_user data: Owner.
  ?cAct abac:accessDay ?d.
  ?d list:in
("Monday" "Tuesday" "Wednesday" "Thursday" "Friday").
  ?cAct platys:occurs_when ?t.
  acadDomain:accessHour time:includes ?t.
} => { ?A a abac:PermittedAction  }.
```

The above policy rule states that the owner can operate the tractor (OBU) from Monday to Friday if the owner is given access permission specified by the property `hasAccessPermission`. The time interval for the owner to access is specified by `accessHour` property like 09:00 AM to 04:00 PM.

C. Scenario 3 - Worker (WBU) Operation Permissions

For this scenario, the FBU and the OBU start communicating with the worker (WBU) only if an instance of WBU `hasOperationPermission` from the owner (HBU) for a specified time period. For example, if the worker (WBU) with `userID` MA1125, has permission from the owner to access FBUs such as Automated_Sprinkler and SoilMoisture_Sensor.

The worker (WBU) automatically gets the data regarding the status of Automated_Sprinkler (FBU) and readings from the SoilMoisture_Sensor (FBU) only for that specified time period. But the worker is not allowed to access previous or future data of the above mentioned FBUs.

V. Conclusion

With the integration of technologies like big data, cyber physical systems (CPS), artificial intelligence, blockchain, etc. farmers can now closely monitor various events and interactions that happen on their smart farms to increase the overall crop yield. A side-effect of this technological integration is an increase in the number of attack surfaces [3]. Specific cyber defensive systems need to be built to protect the smart farm ecosystem.Therefore, in this paper we have created a smart farming ontology to encode farm specific sensors and interactions. We began by creating a smart farm ecosystem architecture that consists of physical entities, digital twins and a representation graph. We categorize the farm equipment and explain in detail various interactions that happen between the owner, workers, sensors, and vehicles that are present on the farm. Using this architecture and interaction model we created a smart farm ontology. Using our base ontology, with a context and platys ontologies we implement Attribute Based Access Control for the farm. We also explained potential use case scenarios and how access control decisions are made using our smart farming ontology.

Acknowledgment

This research was partially supported by a grant received from the National Institute of Standards and Technology (NIST), United States Department of Commerce.

References

[1] Max Roser. Future population growth. *Our world in data*, 2013.

[2] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal. Security and privacy in smart farming: Challenges and opportunities. *IEEE Access*, 8:34564–34584, 2020.

[3] Aida Boghossian et al. Threats to Precision Agriculture. Technical report, U.S. Department of Homeland Security, 2018.

[4] Sjaak Wolfert, Lan Ge, Cor Verdouw, and Marc-Jeroen Bogaardt. Big data in smart farming – a review. *Agricultural Systems*, 153:69 – 80, 2017.

[5] Laxmi S Shabadi and Hemavati B Biradar. Design and Implementation of IoT based Smart Security and Monitoring for Connected Smart Farming. *International Journal of Computer Applications*, 975:8887.

[6] Xin Jin, Ram Krishnan, and Ravi Sandhu. A unified attribute-based access control model covering dac, mac and rbac. volume 7371, pages 41–55, 07 2012.

[7] Callum Eastwood, Laurens Klerkx, Margaret Ayre, and Brian Dela Rue. Managing socio-ethical challenges in the development of smart farming: From a fragmented to a comprehensive approach for responsible research and innovation. *Journal of Agricultural and Environmental Ethics*, 32:741–768, 11 2019.

[8] Spyros Fountas, Giacomo Carli, Claus Sørensen, Zisis Tsiropoulos, Chris Cavalaris, Anna Vatsanidou, B. Liakos, Maurizio Canavari, Jens Wiebensohn, and Bruno Tisseyre. Farm management information systems: Current situation and future perspectives. *Computers and Electronics in Agriculture*, 115:40–50, 07 2015.

[9] UK Shanwad, V.C. Patil, Ghulappa Dasog, CP Mansur, and KC Shashidhar. Global positioning system (gps) in precision agriculture. *Proceedings of Asian GPS conference*, 01 2002.

[10] L. Cancar, David Sanz, Juan Hernández Vega, Jaime Cerro, and Antonio Barrientos. *Precision Humidity and Temperature Measuring in Farming Using Newer Ground Mobile Robots*, volume 252, pages 443–456. 01 2014.

[11] Noor-E Sabiha and Sanzidur Rahman. Environment-smart agriculture and mapping of interactions among environmental factors at the farm level: A directed graph approach. *Sustainability*, 10:1580, 05 2018.

[12] Feng Yang, Kaiyi Wang, Hanyun Han, and Zhong Qiao. A cloud-based digital farm management system for vegetable production process management and quality traceability. *Sustainability*, 10:4007, 11 2018.

[13] Gaoqi Liang, Steven Weller, Junhua Zhao, Fengji Luo, and Z.Y. Dong. The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems*, PP:1–1, 11 2016.

[14] Maria Bartnes, Ali Zand, Gianluca Stringhini, and Richard Kemmerer. Targeted attacks against industrial control systems: Is the power industry prepared? *Proceedings of the ACM Conference on Computer and Communications Security*, 2014:13–22, 11 2014.

[15] Molly M. Jahn et al. Cyber Risk and Security Implications in Smart Agriculture and Food Systems. Available at : https://jahnresearchgroup.webhosting.cals.wisc.edu/wp-content/uploads/sites/223/2019/01/Agricultural-Cyber-Risk-and-Security.pdf (Accessed on: 2019/11/14), 2019.

[16] Daniel Lopez, Maria Uribe, Claudia Santiago, Andrés Torres, Nicolas Guataquira, Stefany Castro, Pantaleone Nespoli, and Felix Gomez Marmol. Shielding iot against cyber-attacks: An event-based approach using siem. *Wireless Communications and Mobile Computing*, 2018, 10 2018.

[17] Tawfik Mudarri, Samer Al-Rabeei, and Samer Abdo. Security fundamentals: Access control models. *Interdisciplinarity in theory and practice*, 08 2015.

[18] Ravi S Sandhu and Pierangela Samarati. Access control: principle and practice. *IEEE communications magazine*, 32(9):40–48, 1994.

[19] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.

[20] Maanak Gupta, James Benson, Farhan Patwa, and Ravi Sandhu. Dynamic groups and attribute-based access control for next-generation smart cars. In *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*, pages 61–72, 2019.

[21] Grigoris Antoniou, Enrico Franconi, and Frank Harmelen. Introduction to semantic web ontology languages. pages 1–21, 01 2005.

[22] Vishal Rathod, Sandeep Narayanan, Sudip Mittal, and Anupam Joshi. Semantically rich, context aware access control for openstack. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pages 460–465. IEEE, 2018.

[23] Maithilee Joshi, Sudip Mittal, Karuna P Joshi, and Tim Finin. Semantically rich, oblivious access control using abac for secure cloud storage. In *2017 IEEE international conference on edge computing (EDGE)*, pages 142–149. IEEE, 2017.

[24] S. Mittal, A. Gupta, K. P. Joshi, C. Pearce, and A. Joshi. A question and answering system for management of cloud service level agreements. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, pages 684–687, June 2017.

[25] Karuna P Joshi, Aditi Gupta, Sudip Mittal, Claudia Pearce, Tim Finin, et al. Alda: Cognitive assistant for legal document analytics. In *2016 AAAI Fall Symposium Series*, 2016.

[26] Horrocks, Ian, Patel-Schneider, Peter F, Boley, Harold, Said Tabet, Said, Grossof, Benjamin, Mike Dean, and Mike. Swrl: A semantic web rule language combining owl and ruleml. *W3C Subm*, 21, 01 2004.

[27] Mike Dean and Guus Schreiber. Owl web ontology language reference: W3c recommendation 10 february 2004. 02 2004.

[28] Rosa Gutierrez, Pradeep Murukannaiah, Nithyananthan Poosamani, Tim Finin, Anupam Joshi, Injong Rhee, and Munindar Singh. Platys: From position to place-oriented mobile computing. *Ai Magazine*, 36:50–, 07 2015.