Using Attribute-Based Access Control for Remote Healthcare Monitoring

Indrakshi Ray*, Bithin Alangot†, Shilpa Nair†, Krishnashree Achuthan†
*Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873
indrakshi.ray@colostate.edu
†Amrita Center for Cybersecurity Systems and Networks
Amrita School of Engineering, Amritapuri
Amrita Vishwa Vidyapeetham, Amrita University, India
bithina@am.amrita.edu, er.shilpa23@gmail.com, krishna@amrita.edu

***Abstract*—Remote Healthcare Monitoring (RHM) IoT infrastructure uses sensors and smartphones to collect vital parameters from patients. These parameters pertaining to medical records are shared with healthcare professionals at geographically distant locations to provide timely medical care. RHM applications deployed on IoT infrastructure must address the issues of security and privacy in a constrained environment. We present our H-Plane framework for RHM and propose the use of the NIST Next Generation Access Control (NGAC) framework for specifying and enforcing access control policies.**

I. Introduction

Remote Healthcare Monitoring (RHM) is a critical IoT infrastructure that uses sensors and smartphones to collect vital parameters such as blood pressure, blood sugar, and heart rate from patients. These medical records are shared with healthcare professionals including doctors, nurses or paramedics who may be at geographically distant locations to help provide timely medical diagnosis and prescription. Through RHM, healthcare personnel are able to support timely care especially for the elderly and disabled who are unable to access medical facilities in case of emergencies. Providing such care remotely at patients' home may also reduce unnecessary need for hospitalizations, re-admissions, and prolonged stay in hospitals resulting in affordable cost of care for common citizens. For example, a patient who has a minor cardiac surgery will need to make periodic visits to the hospital to monitor their health conditions, take necessary precautions for preventative care. These visits can be significantly reduced as part of post-operative care by using sensors that collect vital information regarding the heart such as EEG and ECG data at frequent intervals and automatically alert the care providers of any health anomalies.

The current IoT infrastructure facilitating RHM applications have to deal with issues related with scalability, bandwidth utilization, latency, durability management along with privacy and security concerns due to their constrained design [21]. In order to overcome these limitations, we came up with an IoT framework called Healthcare Plane (H-Plane) [17] that is designed based on the concept of Global Data Plane [14]. The framework was developed to cater to the needs of an indigenously developed RHM application [3] that remotely monitors patients' heart abnormalities. One of the key advantages of H-plane is its ability to effectively overcome most of the constraints that exist in current IoT infrastructure architectures.

However, privacy issues in H-Plane pertaining to RHM are yet to be addressed. Health care data consists of sensitive information the disclosure of which may compromise the privacy of the individual and also cause financial and physical damage to the patient. Further, it is also important to have the access control techniques used for critical care to comply with regulations such as Health Insurance Portability and Accountability Act (HIPAA) [6] that defines mandatory privacy rules to protect sensitive personal identifiable health information. As per HIPAA, the access control should provide fine grained policies to individual users, incorporate accountability, support revocability of privileges, and provide scalability and flexibility.

We need an access control model and enforcement mechanism that provides adequate protection to healthcare data. Most of the commonly used access control techniques such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role Based Access Control (RBAC) and its extensions do not satisfy all the above stated requirements. MAC may be too restrictive, DAC policies may be hard to manage, and RBAC may be inadequate for policies that may span different organizational boundaries. We propose the use of Attribute-Based Access Control (ABAC) for specifying policies over healthcare data. The two emerging standards in ABAC include eXtended Access Control Markup Language (XACML) [16] and NIST Next Generation Access Control (NGAC) framework [5], [4]. XACML is expressive but support for policy management is relatively poor. NIST NGAC, on the other hand, provides a uniform mechanism for expressing and managing policies. Moreover, it separates the access control logic from the operating environment making it easier to use and understand. In addition, policies can be efficiently enforced. Although health care policies have been modeled using XACML, this is the first attempt for modeling policies

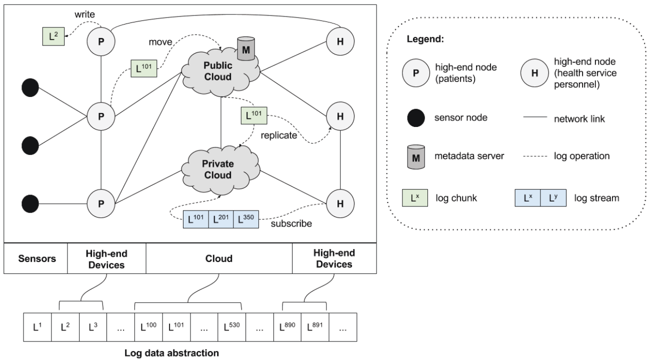

Fig. 1. HPlane Architecture

using NIST NGAC.

The rest of the paper is organized as follows. Section II gives a background on the NIST NGAC and H-Plane. Section III describes some related work in this area. Section IV illustrates our access control model. Section V concludes the paper with pointers to future directions.

II. Background

In this section, we provide an overview of H-Plane and NGAC frameworks. H-Plane enables developers to build complex IoT applications with ease and NGAC helps enforce access control policies in these applications.

A. H-Plane Architecture

H-Plane is a data centric framework that use append-only log data structure for fundamental storage abstraction. The H-Plane framework aggregates the data generated by the wearable sensors attached to the patients and stores it as a stream in an append-only log file at the patient's smartphone which acts as the IoT gateway. The smartphone assesses the data criticality and tags the data before it gets stored in the log file. Subsequently, the log file gets transferred to other edge nodes or to the backend cloud nodes based on its criticality. Though the application gets a single logical view of the log, physically it is divided into different segments to meet the Quality of Service (QoS) requirements. The sensing device, smartphone and cloud forms the IoT infrastructure that helps us to communicate information among the various users. Figure 1 shows the architecture of H-Plane.

We adopt a data centric approach using log abstraction for efficient storage and communication of data across the distributed nodes (high-end sensor device, smartphones, cloud servers). All the devices and user applications are abstracted with a log interface, so any interaction with them happens via a log file. This is similar in spirit to how operating system provides a file abstraction to a device, reading and writing to a device is basically a file operation. The users use the services through a user application which is an Android App that we refer to as Amrita Spandam (AS). Our goal is to provide access control such that users get access to only those services that they are authorized to perform.

B. NIST NGAC

Typically, in a complex application, there are variants of access control policies that are defined and enforced at different levels of application logic. So one must ensure that these different types of policies are adequately enforced to prevent confidentiality and integrity breaches. NIST NGAC is a specification and enforcement mechanism that can supports various types of fine-grained access control policy in the same framework. The framework consists of fixed set of relations and functions between policy elements and a reference mediation to render access control decisions. These relations and functions express and enforce multiple policies while allowing reconciliation of policy conflict under a single access control model. Additionally, the framework is also capable of expressing and enforcing nondiscretionary and discretionary access control policies as the policies are expressed using attributes.

The core policy elements of PM includes (a) authorized users and user attributes, (b) protected objects and object attributes, (c) administrative and non-administrative operations, (d) administrative and non-administrative access rights, and (e) policy classes that organize and distinguish between different kinds of policies expressed and enforced. We use these policy elements to define assignments, associations, privileges, prohibitions and obligations in the context of our application.

III. Related Works

Most works on protecting healthcare data make use of RBAC as the primary model of choice for access control. However, RBAC suffers from its own deficiencies. Firstly, RBAC is not granular expressive enough to accommodate fine-grained access control that depends on factors such as the requester's age, hippa-compatibility, name etc. Secondly, RBAC grants similar access to people having the same roles. In spite of its advantages in scalability and management, RBAC is not considered quite suitable for the medical domain [18].

ABAC model provides a more dynamic and granular approach to access control. Users are allowed to define access control policies using attributes of different entities like subjects, actions, resources and environments. ABAC has been used to secure access to healthcare data [18] using the XACML standard. It has also been used for protecting healthcare objects specified in the form of Fast Health Interoperability Resources (FHIR) [7] [15]. User-Managed Access (UMA) [12] specify protocol standards for user centric access management of stored healthcare resources but it does not provide the essential implementation details of an ABAC based security system on a FHIR server.

RHM concept was initially designed to provide quality and timely medical attention to elderly and disabled people from their home. But with the advancements in hardware as well as software technologies, it is now possible to develop RHM architecture that extends these medical services from remote location to regular patients as well. For example, mPHASIS [9] provides sensor networking, vital sign monitoring and signaling alert messages to the caregiver using smartphones over a heterogeneous network. Similarly there exists RHM architecture called Medical Image Access and Presentation System (MIAPS) which is a web based application that uses the concept of image analysis [20]. But these technologies are still prone to IoT infrastructure issues related to privacy and security, latency, scalability, bandwidth, and data durability management due to the application design. Keeping this in mind, our Amrita Spandanam was designed using H-Plane framework which overcomes these constraints.

The research work such as [13] [1] [2] [8] studied privacy threats that exist in RHM technology and identify various access control model requisites. These access control models should comply with government initiated rules that protect the privacy of patient's health records. The patients' health records are stored as Electronic Health Records (EHRs) or Personal Health Records (PHRs) and processed in clouds. Many privacy preserving frameworks such as [11] [19] [1] [22] have been developed for cloud technology, but often times the cloud technology falls short with respect to latency and flexibility [21]. This has been resolved by H-Plane framework that is developed for Amrita Spandanam. But the privacy issues in H-plane still persists. We wish to define an access control model that will allow us to combine different types of access control policies. Currently, XACML standards are used to design such access control model for various enterprises [10]. So the access control conceptual prototype we constructed is first of a kind that uses the NIST NGAC framework for an IoT infrastructure.

IV. Access Control Model

In this section, we describe our access control model for RHM using NGAC. We first present the healthcare access control requirements using a motivating example and then describe how we model it using NIST NGAC.

A. Motivating Example

We describe a normal and emergency scenario of our application workflow implemented over the H-plane framework. AS ecosystem consists of several users and devices which generate privacy sensitive information. It is important to provide strict access control to protect the data from unauthorized access but still enable a hassle free service.

We have a patient *John* who recently had heart surgery. In order to continuously monitor his health remotely, we are using our AS remote health monitoring system. *John* is attached with a sensing device that monitors his heart rate and sends it to his smartphone via bluetooth. The smartphone that acts as the gateway device runs the application AS. AS does some processing on the data, classifies it as *critical* or *routine* and sends it to the cloud. Healthcare professional running the AS application is able to retrieve the data from the cloud and take remedial action.

The sensor connected to *John* writes the data to the log file stored in his smartphone. The data is processed by the application instance (into which he is authenticated) running locally in order to decide the criticality or detect any anomalies. Only *John* and the associated sensor devices can have read access to his log files. In addition, he can create a new log from his existing logs and delegate access to his logs to other users. When there are no anomalies, the sensor data is written into a normal log which is chunked into smaller segments and moved into the cloud. The applications and the devices are unaware of the log segments as they get a single unified view of the normal log. *John*'s doctor, *Dr. Mark*, subscribes to *John*'s log. *Dr. Mark* can pull the data from the normal log as and when needed.

In case an abnormality is detected, the data coming from the sensor is written into a critical log which is physically a different file. *Dr. Mark* gets subscribed to this critical log so that he can take quick remedial actions. When the abnormal readings leads to an emergency situation, an emergency log is created. This emergency log is created from the recent normal log file and critical log file. *Dr. Mark* can delegate the access to the nurse *Mike* and other doctors to assist him in the situation. Emergency responders should also get access to the emergency logs. Once the emergency is resolved, the authorization given to the responders will be automatically revoked.

Access control is based on the role and other attributes of the user. We now summarize the permissions associated with the various roles in the healthcare domain.

- **Patient** A patient has *read* access to all her log files. A patient can also *delegate* the access she has to any other user on the basis of his attributes.
- **Doctor** A doctor can *read*, *append*, and *write* logs belonging to her patients. A doctor can *delegate* a subset of her privileges to other doctors and/or nurses.
- **Nurse** A nurse can *read* and *append* logs belonging to her patients only during working hours.
- **Healthcare Provider** has *read* access to patient's log data.
- **Emergency Responder** has *read* access to all the patients' log only during the emergency provided the responder and the patient are in the same location. In addition, the emergency responder has *append* and *write* access to a patient's emergency log.
- **Family Member** One or more family member is given *read* access to a patient's log.

In addition, all privileges delegated by the doctor or the patient must be logged. Privileges given to an emergency responder must be revoked once the emergency is over.

B. Policy Elements

We describe our policy elements in this section.

User This is the set of entities who have an account in the AS. For example, *User* = $\{John, Mark, Sara, Clara, Barbara\}$.
User Attributes Each user is associated with a set of attributes. Examples of attributes include *identity*, *role*, and

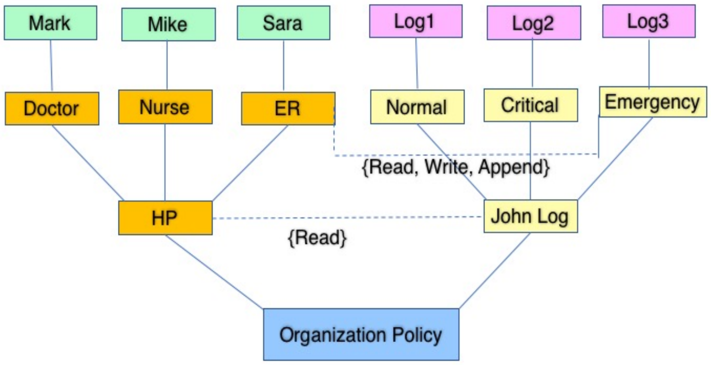

Fig. 2. Partial Access Control Model for Amrita Spandanam

credentials that are important for authorization decisions. *identity* gives the identity by which the user is known in *AS*. *role* determines the role of the user. *role* is an enumerated attribute as shown below.
role = { *Admin, Patient, Doctor, Nurse, Family, Emergency Responder, Healthcare Provider* }. For example, the role attribute of *Barbara* is *Admin* and that of *Mark* is *Doctor*. Credentials of the healthcare provider may also determine her access. Credentials may include *medical degree*, *specialization*, and *experience*. Each of these can be specified as an enumerated attribute. *location* is the location of the user and is once again specified as an enumerated attribute as given below.
location = { *hospital, home* }
Values of some enumerated attributes, such such as *roles* and *medical degree* can also be arranged in a hierarchy that implies that an attribute at the higher level inherits from the attribute at the lower level.

Object These is the set of objects needing protection. Examples include log files generated from various patients.

Object Attributes The object attributes include the *patient_id*, *data_criticality*, and *data_recency*. *patient_id* gives the identity of the patient whose information is contained in the log. *data_criticality* and *data_recency* can be specified as enumerated attributes as shown below.
data_criticality = { *normal, critical, emergency, combined* }
data_recency = { *recent, archived* }
Note that, the enumerated attributes in each of these cases are mutually exclusive.

Operations The set of operations on the objects include *read log*, *write log*, and *append log*. There are also administrative operations that include *create log*, *create user*, *create user attribute*, *change user attribute*, *delete user*, *delegate*, *create object attribute*, *change object attribute*, *grant permission*, and *revoke permission*. Some administrative operations are performed by the role *Admin* whereas others are done by users associated with the objects. For example, *create user*, *delete user*, *create user attribute*, *create object attribute*, *change user attribute*, *grant permission*, and *revoke permission* are done by the role *Admin*. The operation *delegate* is performed by users in some designated roles, including *Patient* and *Doctor*. Some operations are done by the trusted applications. For example, generation of log from sensor data or classifying the log is done automatically by the applications. Thus, *create log* operation is done by the patient's smartphone who also attaches the tags *critical* or *routine*. The tags on the data can also be changed by the role *Doctor*.

We next describe the association between attributes that demonstrates the permissions.

C. Assignments and Associations

The various policy elements are related by assignments and associations.

Assignments These express relationships between (i) users and user attributes, (ii) objects and object attributes, (iii) user attributes and user attributes, (iv) object attributes and object attributes, and (v) user/object attributes and policy classes.

Figure 2 pictorially demonstrates a part of the model of *AS*. The solid lines show the assignment relationship. *Mark* assigned to *Doctor* is an example of user to user attribute assignment. *Doctor*, *Nurse*, *Emergency Responder* (shown as *ER*) are all assigned to *Healthcare Provider* (shown as *HP*). This is an example of user attribute to user attribute assignment. Such assignments indicate the presence of an attribute hierarchy. There are other user attribute to user attribute assignments. For example, *Emergency Responder* is assigned to *Healthcare Provider* – this signifies that the *Emergency Responder* inherits the privileges given *Healthcare Provider*. *HP* and *John Log* are assigned to the policy class *Organizational Policy*. This is an example of an user and object attributes assigned to the policy class called *Organizational Policy*.

Associations These define the access rights assigned to user attributes for performing operations on objects specified by object attributes. The dashed lines show associations. The dashed line between *Healthcare Provider* and *John Log* is labeled with *Read*. This demonstrates that HP can read the objects whose attributes are *John Log*. In addition, the dashed line between *Emergency Responder* and *Emergency* attribute of the log signifies that the *Emergency Responder* can read, write, or append the logs whose attribute is *Emergency*.

The spatio-temporal constraints on permissions are not shown in the figure for the sake of simplicity. This constraint says that *Emergency Responder* can access *Emergency* log only during an emergency and when the patient and the *Emergency Responder* are co-located.

For our motivating example, the set of associations that are related to administrative operations and applications are given below.
SysAssociation = { *(App, create, object), (Admin, create, user), (Admin, create, user attribute), (Admin, change, user attribute), (Admin, delete, user attribute), (Admin, delete, user), (Admin, delete, object attribute), (App, delete, object), (App, change, object attribute), (Admin, change, object attribute)* }

The set of associations that are related to application users are listed below.
UserAssociation = { *(Patient(PatientID), read, log(PatientID)), (Family(PatientID), read, log(PatientID)), (Doctor(PatientID), read, log(PatientID)), (Doctor(PatientID), write, log(PatientID)), (Doctor(PatientID), append, log(PatientID)), (Nurse(PatientID), read, log(PatientID), working_hours), (Nurse(PatientID), append, log(PatientID), working_hours), (Healthcare Provider(PatientID), read, log(PatientID), working_hours), (Emergency Responder, read, log(Emergency,EmergencyPatient), working_hours, Emergency Responder(location) = Emergency Patient(location)), (Emergency Responder, append, log(Emergency,EmergencyPatient), working_hours, Emergency Responder(location) = Emergency Patient(location))*}

V. Conclusion

In this paper, we demonstrate how access control for a medical application that uses an IoT framework can be specified using ABAC. Specifically, we have demonstrated how NIST NGAC can be used for specifying the policies. The policy specification using NIST NGAC is simple and is supported by an efficient implementation. We have shown how the various permissions involving data operations can be modeled. A lot of work remains to be done. We need to demonstrate how policies can be efficiently managed by NGAC. Our future work also includes how the various forms of audit can be supported in the context of NGAC. Our final goal is to incorporate the NGAC enforcement mechanism to the H-plane architecture.

Acknowledgment

The work of Indrakshi Ray was partially supported by NIST Award Number 60NANB16D250, NSF NSF I/UCRC Award Number 1650573, and funding from CableLabs. The views and conclusions contained in this document are those of the authors and should not be automatically interpreted as representing the official policies, either expressed or implied of NIST, NSF, and CableLabs.

References

[1] A. Abbas and S. U. Khan. A Review on the State-of-the-Art Privacy-Preserving Approaches in the E-Health Clouds. *IEEE Journal of Biomedical and Health Informatics*, 18(4):1431–1441, 2014.

[2] O. Boric-Lubecke, X. Gao, E. Yavari, M. Baboli, A. Singh, and V. M. Lubecke. E-healthcare: Remote Monitoring, Privacy, and Security. In *Proc. of IMS*, pages 1–3. IEEE, 2014.

[3] N. Dilraj, K. Rakesh, K. Rahul, and R. Maneesha. A Low Cost Remote Cardiac Monitoring Framework for Rural Regions. In *Proc. of MOBIHEALTH*, 2015.

[4] D. Ferraiolo, R. Chandramouli, V. Hu, and R. Kuhn. A Comparison of Attribute Based Access Control (ABAC) Standards for Data Services. Draft NIST Special Publication 800-178, National Institute of Standards and Technology, 2016.

[5] D. F. Ferraiolo, S. Gavrila, and W. A. Jensen. Policy Machine: Features, Architecture, and Specification. Technical Report NISTIR 7987 Revision 1, National Institute of Standards and Technology, 2015.

[6] C. for Medicare & Medicaid Services et al. The Health Insurance Portability and Accountability Act (HIPAA) of 1996. Online at http://www. cms. hhs. gov/hipaa, 1996.

[7] HL7. Overview - FHIR v1.0.2. https://www.hl7.org/fhir/overview.html, Oct 2015. [Online; Accessed 24-January-2017].

[8] D. Kotz, S. Avancha, and A. Baxi. A Privacy Framework for Mobile Health and Home-Care Systems. In *Proc. of SPIMACS*, pages 1–12. ACM, 2009.

[9] P. Kulkarni and Y. Ozturk. mPHASiS: Mobile Patient Healthcare and Sensor Information System. *Journal of Network and Computer Applications*, 34(1):402–417, 2011.

[10] N. S. Kumar and A. Amalanathan. Unifying the Access Control Mechanism for the Enterprises Using XACML Policy Levels. *International Journal of Information Technology and Computer Science*, 12:82–88, 2015.

[11] M. Li, S. Yu, K. Ren, and W. Lou. Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings. In *Proc. of SecureComm*, pages 89–106. Springer, 2010.

[12] M. P. Machulak, E. L. Maler, D. Catalano, and A. van Moorsel. User-managed Access to Web Resources. In *Proc. of Digital Identity Management*, pages 35–44, New York, NY, USA, Oct 2010. ACM.

[13] M. Meingast, T. Roosta, and S. Sastry. Security and Privacy Issues with Health Care Information Technology. In *Proc. of EMBS*, pages 5453–5458. IEEE, 2006.

[14] N. Mor, B. Zhang, J. Kolb, D. S. Chan, N. Goyal, N. Sun, K. Lutz, E. Allman, J. Wawrzynek, E. A. Lee, et al. Toward a Global Data Infrastructure. *IEEE Internet Computing*, 20(3):54–62, 2016.

[15] S. Mukherjee, I. Ray, I. Ray, T. Ong, S. Hossein, and M. G. Kahn. Attribute Based Access Control for Healthcare Resources. In *Proc. of ABAC@CODASPY*, Scottsdale, AZ, March 2017. ACM.

[16] OASIS. eXtensible Access Control Markup Language (XACML) Version 3.0. Oasis standard, Organization for the Advancement of Structured Information Standards, Jan. 2013. Available from http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

[17] R. K. Pathinarupothi, B. Alangot, M. V. Ramesh, K. Achuthan, and P. V. Rangan. H-Plane: Intelligent Data Management for Mobile Healthcare Applications. In *Proc. of MobiWIS*, pages 283–294. Springer, 2016.

[18] I. Ray, T. C. Ong, I. Ray, and M. G. Kahn. Applying Attribute Based Access Control for Privacy Preserving Health Data Disclosure. In *Proc. of BHI*, pages 1–4, Las Vegas, NV, February 2016. IEEE.

[19] W. Ren, Y. Ren, M.-E. Wu, and C.-J. Lee. A Robust and Flexible Access Control Scheme for Cloud-IoT Paradigm with Application to Remote Mobile Medical Monitoring. In *Proc. of RVSP*, pages 130–133. IEEE, 2015.

[20] H. Shen, D. Ma, Y. Zhao, H. Sun, S. Sun, R. Ye, L. Huang, B. Lang, and Y. Sun. MIAPS: A Web-Based System for Remotely Accessing and Presenting Medical Images. *Computer Methods and Programs in Biomedicine*, 113(1):266–283, 2014.

[21] B. Zhang, N. Mor, J. Kolb, D. S. Chan, K. Lutz, E. Allman, J. Wawrzynek, E. Lee, and J. Kubiatowicz. The Cloud is Not Enough: Saving IoT from the Cloud. In *Proc. of HotCloud 15*, 2015.

[22] J. Zhou, X. Lin, X. Dong, and Z. Cao. PSMPA: Patient Self-Controllable and Multi-Level Privacy-Preserving Cooperative Authentication in Distributedm-Healthcare Cloud Computing System. *IEEE Transactions on Parallel and Distributed Systems*, 26(6):1693–1703, 2015.