Access Control in the Era of Big-Data Driven Models and Simulations

Anne M. Tall, Cliff C. Zou, and Jun Wang
University of Central Florida
Orlando, FL
anne.tall@knights.ucf.edu, changchun.zou@ucf.edu, jun.wang@ucf.edu

ABSTRACT

In today's mobile-first, cloud-enabled world, where simulation-enabled training is designed for use anywhere and from multiple different types of devices, new paradigms are needed to control access to sensitive data. Large, distributed data sets sourced from a wide-variety of sensors require advanced approaches to authorizations and access control (AC). Motivated by large-scale, publicized data breaches and data privacy laws, data protection policies and fine-grained AC mechanisms are an imperative in data intensive simulation systems. Although the public may suffer security incident fatigue, there are significant impacts to corporations and government organizations in the form of settlement fees and senior executive dismissal.

This paper presents an analysis of the challenges to controlling access to big data sets. Implementation guidelines are provided based upon new attribute-based access control (ABAC) standards. Best practices start with AC for the security of large data sets processed by models and simulations (M&S). Currently widely supported eXtensible Access Control Markup Language (XACML) is the predominant framework for big data ABAC. The more recently developed Next Generation Access Control (NGAC) standard addresses additional areas in securing distributed, multi-owner big data sets. We present a comparison and evaluation of standards and technologies for different simulation data protection requirements. A concrete example is included to illustrate the differences. The example scenario is based upon synthetically generated very sensitive health care data combined with less sensitive data. This model data set is accessed by representative groups with a range of trust from highly-trusted roles to general users. The AC security challenges and approaches to mitigate risk are discussed.

ABOUT THE AUTHORS

Anne M. Tall is a student in the Computer Engineering PhD program at University of Central Florida. She received a MSEE from Johns Hopkins University, Baltimore, MD and a BSEE from University of Maryland, College Park, MD. She is currently employed as a principal cybersecurity engineer at The MITRE Corporation. Her research focuses on computer and network security and systems engineering. The author's affiliation with The MITRE Corporation is provided for identification purposes only and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author. Approved by MITRE for Public Release; Distribution Unlimited. Public Release Case Number 19-2162.

Cliff C. Zou is an associate professor in the Department of Computer Science, University of Central Florida. He received a PhD from the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA in 2005. His research interests include computer and network security, computer networking, and performance evaluation. He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE).

Jun Wang is a full professor of computer science and engineering, and director of the Computer Architecture and Storage Systems (CASS) Laboratory at the University of Central Florida, Orlando, FL. He is recipient of the National Science Foundation Early Career Award 2009 and Department of Energy Early Career Principal Investigator Award 2005. He has authored over 120 publications in premier journals such as IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, and leading HPC and systems conferences such as HPDC, EuroSys, ICS, Middleware, FAST, IPDPS.

Access Control in the Era of Big-Data Driven Models and Simulations

Anne M. Tall, Cliff C. Zou, and Jun Wang
University of Central Florida
Orlando, FL
anne.tall@knights.ucf.edu, changchun.zou@ucf.edu, jun.wang@ucf.edu

INTRODUCTION

Damage and misuse of M&S data can impact not only to a single user whose data is disclosed but also have disastrous consequences to the training and mission rehearsal exercises. Data security is protection against unauthorized disclosure, modification or destruction using hardware and software techniques. An initial and key component of a security mechanism is the access control (AC) decision process. That is, AC is a prerequisite to communication integrity, data at rest encryption, and other security services. It is distinct from the identification process in that AC enforces the policies, rules and decisions on who has access to what.

In big data M&S environments, the emphasis is controlling read, write, execute permissions and delegation of access privileges. For big data systems, a different approach to access control is needed than traditional Relational Database Management Systems (RDBMS) schema-based permissions. AC policies need to be applied dynamically since big data systems apply the concept of a schema on job execution, permissions cannot be pre-defined based upon who has access to which rows, columns, or extracted views. In big data systems controlling access needs to be based on attributes associated with the data objects and subjects (users or processes) requesting access.

The design of a secure system that protect against intentional and accidental threats requires a tradeoff between operational impacts, cost and performance. There is currently an emphasis on consolidating, centralizing and interconnecting distributed systems to resolve hard problems. The goal is to reduce data inconsistency and enable application access to data. However, a much more dangerous security problem appears with this trend. As data silos become interconnected, unauthorized data leaks become more likely without well designed, integrated AC features.

This review and analysis presented in this paper focusses on standard-based approaches to apply AC to big data sets where the volume of data is beyond what is traditionally stored on a single computer, (e.g., terabytes and larger), the variety of formats and structure does not lend to easy insertion into a schema (i.e., no-schema data) and the speed or veracity at which the data is generated, communicated, stored and processed is high. In such big data systems, lines of data are appended to the previously stored data rather than modifying previously stored data.

M&S Data Security Requirements

Security services required to protect data used in large scale M&S systems is driven by different uses, such as:

- Single-user versus multi-user
- Service-specific, joint, and multi-national applications
- Distributed versus centralized data storage management and maintenance (Culton, Parkes, & Walrond, 2016)

These variations impact the location where access control decisions are made to enforce the AC policy. The M&S data formats and communication protocols at the application and presentation layers do not have any generally-accepted construct for labeling the sensitivity of data. Although the idea of applying sensitivity labels in the form of meta-data tags has previously been proposed, a method to do this in a highly trustworthy manner has not gained wide acceptance or use at this time.

Data access patterns also drive the sensitivity of the data. Linking, combing and extracting data can enable the derivation of more highly sensitive information. Data used in Operation Blended Warrior is an example of complex AC interoperability, (Goodman, 2017). The requirement to control access to multi-national, distributed, and full spectrum data including health, financial, and weapons systems specifications illustrates the challenges.

In this paper the example use-case of a health-care large data set data is used to further illustrate implementation of AC in a standard manner. Health-care data is subject to a number of national and international regulatory requirements. In the United States, the Health Insurance Portability and Accountability Act (HIPAA) and Health Information Technology for Economic and Clinical Health Act (HITECH).[1] These laws and their supporting policies address the use and disclosure of individuals' health information by covered providers.

AC Models

AC models were established early in computer system development as part of operating systems design and have evolved over time to address a wide range of use cases, including application-specific, multi-users and networked systems. An AC model provides a logical connection between AC rules and the mechanisms used to implement those rules, (Bertino, Ghinita, & Kamra, 2010), (Harrison, Ruzzo, & Ullman, 1976).

Typically, AC models are implemented in a layered manner within a computer-communications systems. The operating system and many application servers such as SharePoint typically provide AC services based upon a discretionary access control model (DAC) where users can grant access to others by configuring file permissions. However, in many domains, especially for specialized models and simulations, who-has-access-to-what is tightly controlled in a centralized manner through the assignment of roles in a Role Based Access Control (RBAC) or Mandatory Access Control (MAC) model. This ensures, for example, that a database application enforces strictly defined, system enforced roles and responsibilities. To control access at a fine grain level, at the data schema or block level, a high-fidelity model is needed. The Attribute Based Access Control (ABAC) model uses metadata tags or attributes to achieve this level of control. The models are not mutually exclusive, in that various versions of the models are used in different components of the networked computer system environment. This concept of overlaying AC models and the associated implementing technology within the M&S environment is highlighted in Figure 1.

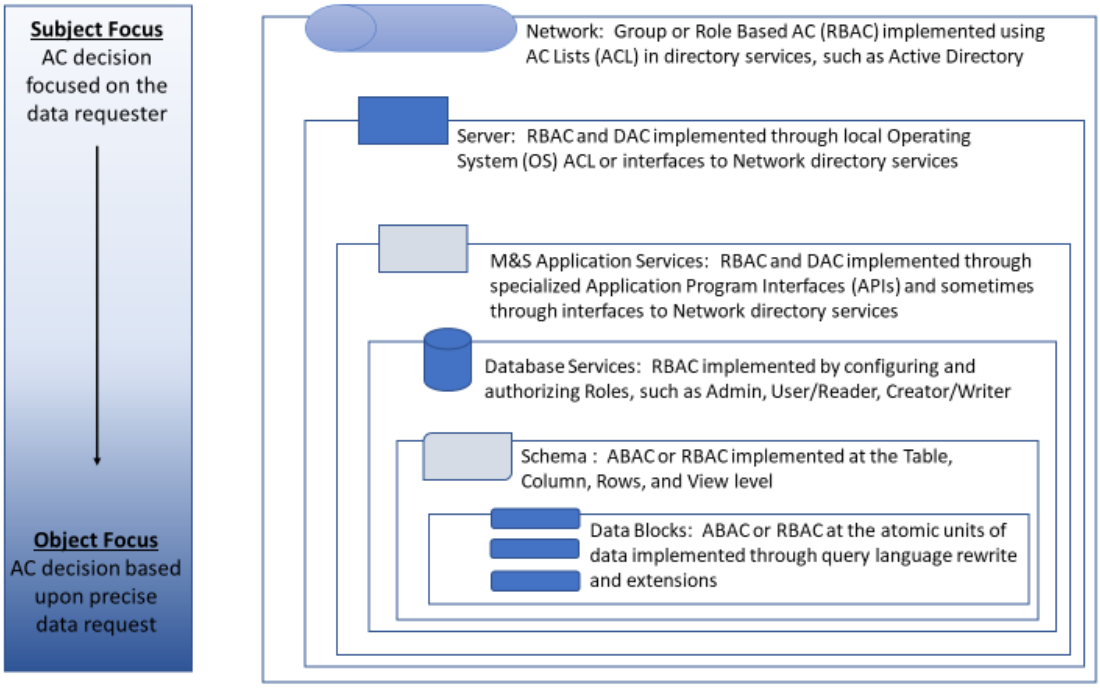

Figure 1: Layered Implementation of AC - layers where AC models are implemented in a M&S environment span from the network to the data block level, with the AC policy focusing more on the subject requesting access at the higher levels to the particular data object that is the target for access at the core data store level.

[1] https://www.hhs.gov/hipaa/for-professionals/index.html

The widely used access control list (ACL) method focusses on listing all authorized subjects, assigning subjects to groups and granting group access to a list of objects. However, these lists are becoming rather large with the current explosion of data and access requests coming from not only human-users, but many processes and Non-Person Entities (NPEs) such as medical sensors and other Internet-of-Things (IoT) devices. To achieve a high-fidelity data block level AC, a different technique is needed. ABAC is a viewed as a method to move away from list-based AC information to enforcement of policy-rules based upon subject and object attributes or characteristics. Enforcement is based upon determining if the subject have the required attributes to access an object with certain attributes, (Hu et al., 2014), (Ferraiolo, Chandramouli, Hu, & Kuhn, 2016). With the next generation data control trend being based upon ABAC, the following sections focus on this AC model. The two primary standards for defining ABAC rules for AC are XACML and the relatively recently defined NGAC, summarized in Table 1.

Table 1: XACML and NGAC Applicability - both XACML and NGAC are focused on defining attribute-based AC control policy enforcement in a standard, interoperable manner.

Standards / Technology	**Applicability**
XACML - eXtensible Access Control Markup Language[2]	An XML-based specification language to express the security policies in terms of rules and the architecture for the access control process
NGAC - Next Generation Access Control[3]	A framework that defines AC in terms of data abstractions and functions based upon attributes associated with users, processes and objects

XACML is the predominant standard for AC rule definition. It is an OASIS standard originally published in 2001 and currently at version 3.0. It is the de facto standard for fine-grained ABAC. XACML defines three parts of an AC system: a policy language, request/response scheme, and an architecture. The policy language defines how to describe authorization constraints in an XML-based structure. The request/response scheme describes the protocol to send authorization requests and receive authorization permission decisions. The architecture contains three main components: enforcement, decisions, and management, as well as several supporting functions, information storage and retrieval. Specifically, the Policy Enforcement Point (PEP), the Policy Decision Point (PDP), Policy Information Point (PIP), Policy Retrieval Point (PRP), and Policy Administration Point (PAP). The core of the architecture is the PIP which loads policies (in XML-format) from the PRP and evaluates the authorization request intercepted by the PEP against those policies using additional information from the PIP when appropriate. The PDP passes the permission request response to the PEP which then permits/denies access to the requesting user/subject. The architecture describes decoupling the authorization decision into logical components that can be incorporated within the appropriate component, exposed at the necessary interface within the overall system architecture (e.g., presentation tier, web-application tier, data storage tier). This enables consistent enforcement across the multiple layers.

NGAC is centered on configuration of relations. AC policies are enumerated based upon associative expressions. NGAC defines the expressions of the policy mode using four types of relationship configurations: Assignment, Associate (derive), Prohibit, and Obligations (dynamic). In NGAC- Generic Operations and Data Structures (GOADS) policies are expressed using the Z formal specification mathematical notation (ISO/IEC 13568:2002 - ZNOT). This is intended to enable validation and management of complex policies and relationships.

EVALUATION OF AC STANDARDS

Fundamentally all AC decisions are based upon the making a grant or deny decision for a subject, (requesting user or process), to take an action on an object, (data or process resources). Complexity is introduced with the dynamic and non-traditional characteristics of the components in this decision. The requestor maybe a Non-Person Entity (NPE) and the decision to grant access to data maybe based upon what data that process has previously gathered (e.g., separation of duty issues). Environmental conditions such as time of day, location of the requestor, and legitimate relationships between data owners and requestors may all also be considerations. As a result, the AC mechanism for large, sensitive M&S data sets sensitivities need to have capabilities/features to handle these complexities.

[2] https://www.oasis-open.org/committees/xacml/
[3] https://webstore.ansi.org/standards/incits/incits4992018

Two key guiding principles that fundamental to the AC design are:

1. The AC model should be expressed in terms of a logical data model, e.g., a relational data model used in a RDBMS or relation attribute tuples used in large data sets
2. Name-based and content-view-based AC are both required, access decisions are based upon at least the subject and object.

Summarized in Table 2 below, and further detailed in this section, the two leading standards that implement RBAC and ABAC model policies, XACML and the more recently developed NGAC were evaluated against five key criteria. This is based upon the National Institute of Standards and Technology (NIST) guidelines and publications [4][5]. Although the NGAC standard provides potentially more robust technical services, especially with administration and management of AC policies both from the subject and object perspective, XACML has wider support and been adopted into more implementations. The technical benefits of shifting to NGAC will need to be clearly appreciated to transition an installed base to a new construct. However, the data integration objectives and the data volumes are continuing to grow, so an approach better suited to the new era of big data management maybe timely for adoption, if the available supported products can be offered at enticing price points with acceptable transition strategies.

Table 2: XACML and NGAC Comparison - In the primary areas of evaluation described in the previous section, NGAC provides advantages, however XACMLs continued wide spread adoption and support may limit NGAC adoption.

Evaluation Criteria	**XACML**	**NGAC**
Security	Complexity makes deployment in a secure manner challenging, however increased use and experience may mitigate this risk	Ensuring complete secure deployment is technically challenging and with limited technical implementation references and community expertise
Policy Expression and Support	Supports some decentralized policy administration by an external delegation model	Objects/resources can be represented with minimal metadata, Weaker at handling environmental attributes and rules with a wide variety of attribute types, Supports history-based policies, and user-independent processes
Operational Efficiency	Less efficient, for each decision, policy loaded into memory and then evaluated	More efficient approach, where policy is loaded into memory at PDP initialization and updated as needed, enables linear scaling
Policy and Attribute Administration and Management	AC rule expression can become very complex, the standard does not define a methodology to review and verify permissions granted by subject or object and address delegations, overrides, and revocations. Metadata must be associated with every object/resource Strong at handling with a variety of attribute types within a trusted domain Does not address efficient policy review	Standard interface for attribute and policy administration Supports efficient algorithms for object and user review NGAC designed more efficiently in policy organization and execution Designed to handle more dynamic conditions By representing the process in the AC decision, NGAC provides greater policy flexibility and support NGAC provides more efficient policy review
Vendor Neutrality - Vendor Lock-In, Separation from Proprietary Operating Systems (OS)	In many implementations of XACML, the PEP is dependent upon the underlying OS	The NGAC definition enables near complete independence from the OS

Security

The most important area for consideration is the overall security afforded by the AC service. Security is achieved through reliable services that protect against threats to the AC services. The critical security requirements include:

- Safety property – ensuring that the execution of a sequence of manipulation operations does not result in access being granted out of compliance with the access control policy
- Data leakage / loss prevention – controlling the use of sensitive data within an organization to only those authorized and with a need-to-know by closely tracking/auditing sensitive data use
- Conflicts of interest management – identifying and preventing permission and access to data associated with organizations with competing or conflicting goals, activities, or objectives.
- Query privacy / Oblivious Transfer (OT) – avoiding the ability to infer information based upon the data access requests
- Bypass prevention - ensuring AC mechanisms, especially when implemented in client-side systems, cannot circumvent the implementation of the AC services.

Risk estimation to make security tradeoffs is an important consideration in the design. This is achieved in highly trustworthy approaches by using a Secure Context, information related to the execution of the data query is encapsulated to reduce risk. For example, for certain functions (e.g., select, insert, delete), argument elements are added (logically "AND"-ed using a WHERE function) to limit the returned data. However, query modification can have drawbacks that affect the correctness of the results and may also negatively impact scalability.

The same level of security should be enforced no matter how the data is accessed. This is achieved in some implementations submitting all service requests (e.g., read, search, write) through a gateway. In some implementations the SQL query statement is rewritten at the gateway to incorporate AC features. A single point of access can also enable single entry and synchronization of AC policies across distributed data stores logically located behind the gateway.

Policy Expression

Policy expression has to do with the scope and type of AC policy model supported. For example, the ability to support dynamic separation of duty. Flexibility in the expression and enforcement of permissions allows for: deny overrides, permit overrides, and first applicable based on order of authorization and/or policy processing. As described in the previous section, there are a wide variety of AC models that apply to different use cases and layers within a M&S computer, communication system environment, so as a result, not all technical approaches lend themselves to cover all conditions. However, ideally the selected approach allows for sufficient flexibility for expressing a wide range of models in both a centralized and decentralized manner.

Operational Efficiency – Performance Impact

The processing overhead for AC decisions can be significant. The algorithm selected for identifying and applying the applicable policy ideally supports linear (rather than exponentiation) scaling as the number subjects and objects increases. Organizing the policies in a manner that allows for loading in memory only the subset applicable to a decision request helps achieve performance requirements. Selecting the policies applicable to the target environment, includes addressing combinations from different authorities, overrides, and handling of conflicts. Policies and the subject/object/action attributes could be organized in various ways such as in a hierarchical graph format. However, the organization can directly impact the performance for loading into memory and processing the AC decision request.

The other factor in considering operational efficiency is if the system architecture model enables externalization of AC policy authorization decisions from within an application to a networked resource. By externalizing the AC decision, the AC service could be potentially used by multiple applications, thus potentially reducing management/maintenance overhead. However, the responsiveness of the external service would need to be scaled to ensure performance requirements. The representation of requests from multiple applications needs to be consistent or standardized to ensure consistent execution.

Performance can be analyzed during three different phases of the AC decision process:

1. Loading AC policies into memory for processing
2. Finding the appropriate policy applicable to the access decision under evaluation
3. Computing or processing the policy to output an access decision

Performance in these three areas is directly driven by the ability to concisely and flexibly define policies so they can be succinctly segmented into manageable portions that can be efficiently processed. Verbose expressive languages to describe policies can be contrary to this goal. Concisely expressed notation can achieve more efficiencies.

Policy and Attribute Management

The complexity of maintaining and checking the integrity of policies and attributes can be a differentiator in the selected approach to AC. A user friendly, intuitive design can increase security by helping to avoid configuration errors. The treatment of attributes can become an unwieldy, confusing bottleneck in the AC decision process, creating information management challenges larger that the dataset/database the AC process is intended to protect.
The difficulty of the AC policy and attribute management approach is addressed through:

- Support for administrative review and integrity checking of AC policies and attributes assigned to subjects, objects and actions.
- Ability to discover resources by reviewing the granted access privileges for subjects and objects
- Fine grain administrative management and controls of who can (create/modify) administer policies, including the ability to delegate and inherit AC administration responsibilities across related targets (i.e., the policies; policy sets that apply to the subjects, objects, actions and environment within the elements of the AC schema)

Standards can leave the area of policy and attribute management as an implementation-specific decision. However, to reconcile access privileges across multiple authorization authorities/officials in a distributed environment, a means to harmonize policies and attributes needs to be conducted in a standard agreed upon manner.

Vendor Neutrality Versus Vendor Lock-in

Vendor neutrality versus vendor lock-in is directly related to the completeness of the applicable standards. Specifically, this issue has been associated with the Policy Enforcement Point (PEP). Standard interfaces to multiple PEPs from multiple applications provides greater flexibility. One-to-one interface from an application to a single PEP constrains a solution to a traditional operating system access control model.

Also, the administration of policies across a distributed environment with multiple authorities needs to be extensible in a standard way to avoid vendor lock-in. For example, workflow, calendar, and records management applications may need interfaces to several PEPs. Tight coupling to an operating system limits the ability to use an integrated AC service across multiple networked applications.

POLICY EXPRESSION

In this section we compare the expression of an example medical data AC policy using the XACML and NGAC standards. The focus is medical healthcare data with defined security codes and a sample community of users with assigned attributes.

Representative healthcare data can be generated using Synthea™ [4] , (Walonoski et al., 2018), synthetic patient medical data generator. Data is generated in the Health Level Seven International (HL7) Fast Healthcare Interoperability Resources (FHIR) specification format.[5] In this example policy scenario there are users in different roles with different levels of trustworthiness and need-to-know. Specifically, Doctors who have a relationship with a Patient have access to all the data associated with the user identifier (UUID). However, Researchers have access to the data only after it has been sanitized, that is meet an obligation for redaction of the certain data fields. HIPAA guidelines specify that de-identification technique mask 16 direct identifiers (e.g., names, email addresses, social security numbers, etc.) and

[4] https://github.com/synthetichealth/synthea/wiki
[5] https://www.hl7.org/fhir/security.html#binding

that quasi-identifiers be generalized (e.g., remove specific date from a birth date and providing only the month and year). The overall challenge with this scenario is controlling access sensitive healthcare data while also making it available to researchers for M&S applications. The key requirements for the policy are:

- Make the healthcare data available to Doctors in a legitimate relationship with the Patient
- Make redacted healthcare data available to Researchers, with the agreement the data will not be retained or reused.

Applying an Attribute-Based Access Control (ABAC) model in this example, a user requests to perform operations (read, write) on objects, EHR data. That user's access request is granted or denied based on a set of access control policies that are specified in terms of attributes and conditions. The attributes include security tags, environment conditions, and user and object characteristics. Attributes are input to the access control policies decision process that determines the operations a user may perform on a Resource (in FHIR) or object (in ABAC). In this example, we focus on using an attribute to specify that the identified data (object/resource) is not to be further disclosed without explicit consent from the patient. Core security labels defined in FHIR that could be used for example are:

- Context of Use, Purpose of Use: HRESCH (Health Care Research) and PATADMIN (Patient Administration)
- Data Sensitivity, Confidentiality Code: U (Unrestricted) and R (Restricted)
- Control of Flow: DELAU (Delete After Use) and NOREUSE (Do Not Re-Use)
- Value Set Obligation Policy: MASK, REDACT

As demonstrated in the Analytics on eXtremely Large European (AXLE) data project, (Meijer, Gaba, & Havinga, 2014), the data request response project includes processes that apply security labels based upon labeling rules, then make the access policy decision, and meet any required processing obligations before providing the result to the requester. This is illustrated in the Figure 2 below.

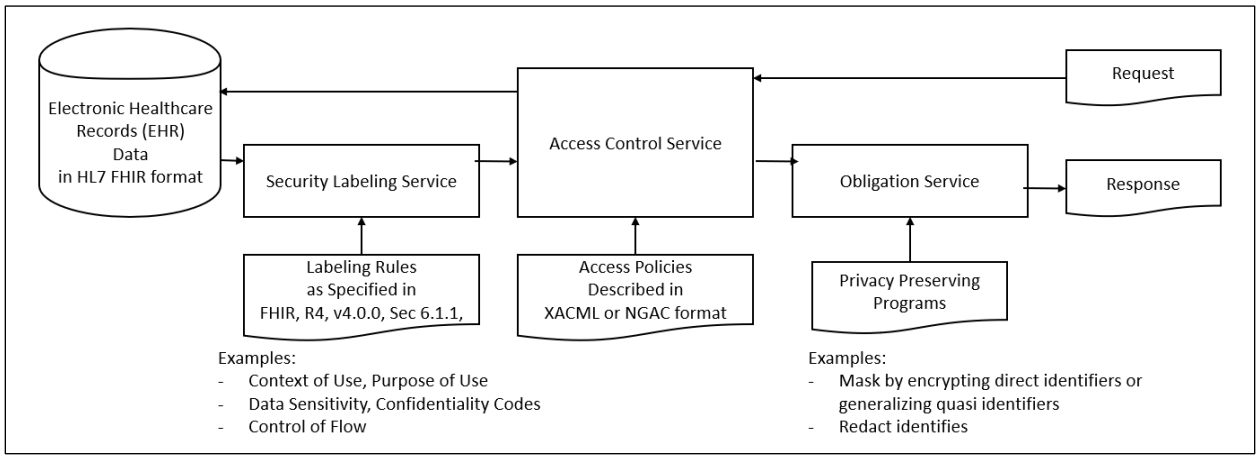

Figure 2: AC Policy Implementation Process - the request and response in Electronic Healthcare Records (EHR) data provisioning includes an AC service that implements access policies as well as security labeling and obligation services.

The ability to express the policy in XACML and NGAC formats was investigated using overarching ABAC guidance from NCCCoE – NIST, (Fisher et al., 2017), the OASIS XACML standard, the NGAC standard and several reference implementations. Example policies for medical data scenario are highlighted in the following sections.

XACML Example Policy Scenario Implementation

Several open source and commercial XACML reference implementations are available that support generation of XACML files from input policies. These provides a starting point options for application developers to incorporate ABAC features. For example, AuthzForce[6] is an open source implementation and Security Policy Tool[7] is a commercial implementation with a free trial version. The complete XACML file generated to fully implement the

[6] https://authzforce.ow2.org/
[7] https://securitypolicytool.com/

policy is too long to incorporate in this paper, however an example of the XACML rule based upon the matching attributes associated with the doctor is shown Figure 3 below:

```
<Rule Effect="Permit" RuleId="rule_1">
  <Target>
   <AnyOf>
    <AllOf>
     <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:http://www.w3.org/2001/xmlschema#string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Doctor</AttributeValue>
      <AttributeDesignator Category="urn:oasis:names:tc:xacml:3.0::subjectcategory:accesssubject"
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Role" DataType="http://www.w3.org/2001/XMLSchema#string"
MustBePresent="true"></AttributeDesignator>
     </Match>
     <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:http://www.w3.org/2001/xmlschema#string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">01-19d82286</AttributeValue>
      <AttributeDesignator Category="urn:oasis:names:tc:xacml:3.0::subjectcategory:accesssubject"
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:Legitimate Relationships"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"></AttributeDesignator>
     </Match>
     <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:http://www.w3.org/2001/xmlschema#string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Any Value</AttributeValue>
      <AttributeDesignator Category="urn:oasis:names:tc:xacml:3.0::attributecategory:resource"
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Resource Type" DataType="http://www.w3.org/2001/XMLSchema#string"
MustBePresent="true"></AttributeDesignator>
     </Match>
     <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:http://www.w3.org/2001/xmlschema#string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">01-19d82286</AttributeValue>
      <AttributeDesignator Category="urn:oasis:names:tc:xacml:3.0::attributecategory:resource"
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Single UUID" DataType="http://www.w3.org/2001/XMLSchema#string"
MustBePresent="true"></AttributeDesignator>
     </Match>
…. several additional match requirements were omitted for brevity …
     </Match>
    </AllOf>
   </AnyOf>
  </Target>
 </Rule>
```

Figure 3: XACML Policy Expression - an example subset of the policy for controlling access to medical records to only doctors with a Legitimate Relationship with a patient is shown. The expression of the complete XACML generated by the Security Policy Tool is several pages long.

The XACML standard includes standard method for tagging a wide variety of attribute values and matching, comparison and evaluation functions above. Although the XACML format is verbose, it is readable by developers and commonly used in a number of applications.

NGAC Example Policy Scenario Implementation

The guidance in the NGAC Functional Architecture (FA) specification uses a diagram to illustrate the policies and relationships. Rule generation is further specified in the NGAC Generic Operations and Data Structures (GOADS) using upon objection relationship notation as tuples. An open source reference implementation for NGAC is the NIST Policy Machine - Harmonia Project.[8] An example figure based upon the medical data scenario is depicted in Figure 4 below.

[8] https://github.com/PM-Master/Harmonia-1.6

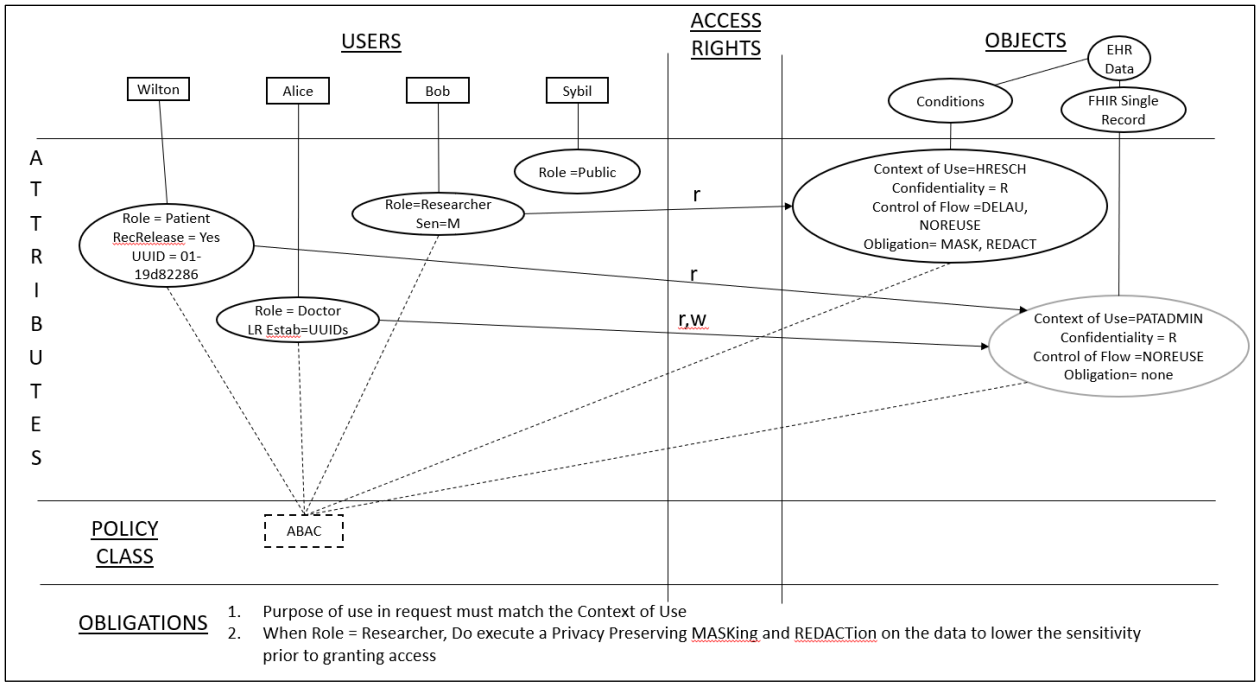

Figure 4: Example NGAC Assignment and Association Graph: The graph illustrates the derived privileges for the example ABAC scenario, which are expressed in tuples as: (Wilton, r, FHIR Single Record), (Alice, r, FHIR Single Record), (Alice, w, FHIR Single record), (Bob, r, Conditions)

CONCLUSION

The protection of big M&S data sets requires an access control solution that is extensible to distributed systems, managed by multiple authorities, and based upon mature standards. XACML is the currently the most widely used standard to implement ABAC, the leading approach for schema on search big data sets. NGAC provides many technical advantages to manage the complexities associated with a large, complex set of attributes for the large number of subjects and objects in a big data environment. An integrated AC approach at the network, system, and data storage level will most likely require applying several AC models as the AC focus shifts from controlling subjects (users) to controlling access to fine grain data objects. XACML and NGAC based reference implementations and products are available to address these challenges. A researched, thoughtful design that applies a standards-based approach would position an M&S system to build upon progress in this domain and enable extending the security coverage as the data sets continue to grow.

ACKNOWLEDGEMENT

This work is supported by the National Science Foundation under grant DGE-1723587.

REFERENCES

Bertino, E., Ghinita, G., & Kamra, A. (2011). Access Control for Databases: Concepts and Systems, *Foundations and Trends® in Databases. NOW Publishers*, vol 3, no. 1-2, pp 1-148. DOI=http://dx.doi.org/10.1561/1900000014

Culton, T., Parkes, D., & Walrond, T. (13 October 2016). Future Construct/Architecture for Modeling and Simulation Support to Joint and Collective Training Across the Continuum of Military Operations, *North Atlantic Treaty Organization (NATO) Science and Technology Organization (STO)*, STO-MP-MSG-143, ISBN 978-92-837-2060-7, from: https://www.sto.nato.int/publications/STO%20Meeting%20Proceedings/STO-MP-MSG-143/MP-MSG-143-20.pdf

Ferraiolo, D., Chandramouli, R., Hu, V., & Kuhn, R. (October 2016). A Comparison of Attributed Based Access Control (ABAC) Standards for Data Service Applications, Special Publication 800-178, *National Institute of Standards and Technology (NIST)* from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-178.pdf

Ferraiolo, D., Feldman, L., & Witte, G. (November 2016). Exploring the Next Generation of Access Control Methodologies, *National Institute of Standards and Technology (NIST)*, from: https://www.nist.gov/publications/exploring-next-generation-access-control-methodologies

Fisher, B., Brickman, N., Burden, P., Jha, S., Johnson, B., Keller, A., Kolovos, T., Umarji, S., & Weeks, S. (2017). Attribute Based Access Control NIST SP 1800-3 Practice Guide, *National Institute of Standards and Technology (NIST)* from: https://www.nccoe.nist.gov/library/attribute-based-access-control-nist-sp-1800-3-practice-guide

Goodman, L. (27 November 2017). Operation Blended Warrior Grey Book, Live, Virtual and Constructive Capabilities, *National Training and Simulation Association (NTSA) and 2017 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*. from https://docplayer.net/76551131-Blended-warrior-grey-book.html

Harrison, M., Ruzzo, W., & Ullman, J. (1976). Protection in Operating Systems, *Commun. ACM* 19, 8, pages 461-471. DOI=http://dx.doi.org/10.1145/360303.360333

Hu, V., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., & Scarfone, K. (January 2014). Guide to Attribute Based Access Control (ABAC) Definition and Considerations, Special Publication 800-162, *National Institute of Standards and Technology (NIST)* from https://csrc.nist.gov/publications/detail/sp/800-162/final

Meijer, H., Gaba, A., & Havinga, Y. (January 2014). Privacy-Aware Analytics on Healthcare Data, *Porta Vita Networked Health*, Analytics on Extremely Large European Data (AXLE) Project, from http://gforge.hl7.org/gf/download/docmanfileversion/7751/11339/AXLE_HL7sec_slides.pdf

Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., & McLachlan, S. (March 2018). Synthea: An Approach, Method, and Software Mechanism for Generating Synthetic Patients and the Synthetic Electronic Health Care Record, *Journal of the American Medical Informatics Association*, Volume 25, Issue 3, Pages 230-238, from https://doi.org/10.1093/jamia/ocx079