

Semantic Attribute Based Access Control

by

Christian Johansen

(UiO)

Table of Contents

1. Abstract.....	1
2. Security and Privacy through Fine-grained access control.....	2
2.1 ABAC in Health and Home Care	2
2.2 Dynamic and Contextual ABAC	2
2.3 Semantic technologies for access control	3
2.4 Query-based Access Control	3
3. Some practical systems for ABAC.....	4
3.1 XACML 3.0 observations	4
3.2 What can we do with XACML	5
3.3 Notes about writing XACML policies	5
3.4 XACML policies implementations	6
4. Research relevant for ABAC.....	6
4.1 Logic based approaches relevant to ABAC	6
4.2 Sticky policies	6
4.3 Attribute Based Encryption (ABE)	6
4.4 Dynamic ABAC or Usage Control (UCON)	7

1. Abstract

The ultimate goal is to empower the other partners to take up the state-of-the-art in security and privacy technologies into their pilots and software. UiO will also conduct research to provide solutions to already identified problems in: semantic technologies, dynamic access control, and privacy-aware distribution of medical data of different granularity and necessity for each requesting health service provider.

Planned internal deliverables are:

- Software modules and software integration, as well as
- Recommendation reports for existing state-of-the-art technologies, including comparisons wrt. requirements derived together with the pilots, and
- Tutorials and manuals for further industry take-up.
- Research papers and reports together with prototype implementations of the proposed research results.
- Testing and usability evaluations of the research results are expected to be done together with the other partners involved in the pilots.

2. Security and Privacy through Fine-grained access control

Our work will be split into three main tasks, and two additional research explorations:

- 2.1. ABAC in Health and Home care (experimental development)
- 2.2. Dynamic and Contextual ABAC (basic research)
- 2.3. Semantic technologies for ABAC (experimental development)
- 2.4. Query-based Access Control (basic research)
- 2.5. Attribute-based Encryption systems (basic research)

Therefore, this Building Block deals with the following topics: security and privacy of access control for sensitive data, semantic technologies, privacy-aware queries (access) on big sensitive data, and ultimately semantic ABAC with dynamic attributes (SABAC). We will also investigate how useful for the project the Attribute-based Encryption (AEC) can be, as a counterpart alternative to ABAC based on encryption techniques, instead of access control techniques.

2.1 ABAC in Health and Home Care

Attribute-based Access Control (ABAC) is the successor of Role-based AC where both resources and subjects have attributes, and a set of attributes can be understood as defining a role. ABAC has reached the maturity of OASI standards with XACML 3.0 and SAML 2.0 (including profiles specific for health-care) with existing tools like open-source Balana or PicketBox from RedHat JBoss or proprietary engines like from Axiomatics.

However, little adoption can be seen in the Health & Home care IT solutions in Europe.¹ If in other industries the role-based approach can be enough, for medical data and processes the ABAC, and more granular extensions of it, are desired due to the highly sensitive and private nature of the information being accessed and the collaborative nature of the work.

Examples: ABAC can handle non-trivial access policies like for collaborative access control, needed in eHospitals, where multiple subjects should be involved, with varying attributes and roles. A classic example is when the Doctor needs to be present (logged in) in order for the Nurse to perform a procedure. Detailed auditing of health-care processes (like administering medicines, preparing operation rooms, home visits) can be done using the notion of obligations in which the decision-point instructs the enforcement-point to first perform some audit/logging actions before granting or denying access.

Our objectives are to establish a strong ABAC foundation for the industry partners to be used within the first phase of the project and tested during the pilots. This task will work towards helping scale current care technologies to the more complex access policies identified by our pilot partners.

2.2 Dynamic and Contextual ABAC

Temporary aspects of access control cannot be handled by ABAC, therefore the recent addition of dynamic attributes and continuous monitoring (which was also termed Usage Control [Park&Sandhu 2004, Martinelli et al. 2010]). In a dynamic and changing working environment like home care or emergency hospitals, it is often that one subject needs to take over a process of another. In this case the attributes of the substitute need to be altered temporarily to enable her to perform the respective duty under the same access rights as the person being replaced. But these changes are temporary; e.g., just for one hour, after which the entrance card into the elderly home would no longer work, or viewing the patient record is allowed during an emergency process to the

¹ More adoption of ABAC can be seen in USA health systems.

doctor on duty, but after that the access becomes again restricted to only the personal doctor.

Our objectives are to introduce dynamic attributes to enable access decisions to be done automatically, so to allow easy integration with tools like those performing optimization of care processes or the semantic reasoners. Moreover, we want to bring the existing research to a stage ready to be taken up by industry. This will provide adaptive policies and dynamic access based, e.g., on the health task undertaken, time, or location of the resources and subjects. This would streamline the health-care work processes at no expense to the privacy of the patients.

2.3 Semantic technologies for access control

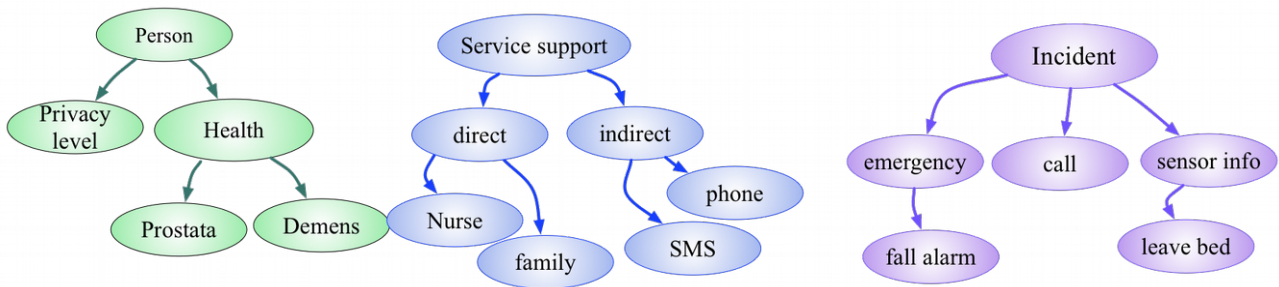
A semantic representation of existing services will be created by translating the static service environment of partners into a machine readable representation, including security and privacy requirements.

Examples of how semantic service descriptions and related ontologies would allow for more complex requirement definitions are:

Eg.1: A person leaving bed, having dementia, needs a nurse.

Eg.2: A person leaving bed, having prostata problems, does not need help if in bed within 30 min, no information should leave the house.

Eg.3: A fall alarm will need support by emergency services, regardless of the privacy/access level.



The envisaged output is a semantic representation of services from the pilots, which will be added to the access control policies developed and implemented in the other tasks. This will form our Semantic ABAC model.

2.4 Query-based Access Control

With increased digitalization of health records and eRegistries (e.g., consider the new DNA banks) and smart homes producing additional environmental data, the health sector enters into the big data era. But in health we cannot grant access to someone for the whole data set, nor for whole records. Instead one often wants to grant access only to those pieces of the data that are needed to provide the respective health service. Even more, one does not want to allow access to data at all (e.g. do not give away not even a single gene sequence), but only grant rights to use some "queries" on the data. The an answer is returned, but not the data, and the health service provider gets what it needs to provide the service, whereas the health records get more confidentiality and privacy protection.

The objective of this work is to investigate the recent advances in query-based information exchange focusing on applying them to provide granular access control to health records. This work aims to provide privacy to the patient data while contributing to making more effective and targeted care processes.

3. Some practical systems for ABAC

3.1 XACML 3.0 observations

- XACML² stands for “eXtensible Access Control Markup Language” and is developed as a standard by the OASIS standardization organization³ (with members of this committee being IBM, RedHat, Oracle, Microsoft, Cisco, Boeing and some 5 more). The 3.0 version was released in 2010 with latest release from 2013.
- A related cousin is [SAML 2.0](#) standing for “Security Assertion Markup Language” and developed also by [OASIS](#), with the release already in 2005. This one is used as a protocol for web authentication to transmit security assertions between domains to achieve things like single sign-on. The close [integration of SAML with XACML](#) is of interests and maybe we should use it.
- There are **subclasses** of the above two specially designed for **Healthcare**, see:
 - [Cross-Enterprise Security and Privacy Authorization \(XSPA\) Profile of Security Assertion Markup Language \(SAML\) for Healthcare v1.0](#)
 - [Cross-Enterprise Security and Privacy Authorization \(XSPA\) Profile of XACML v2.0 for Healthcare v1.0](#)
- XACML is developed for ABAC – Attribute-based Access Control, and thus incorporates RBAC (Role-based) as a special case.
 - XACML engines are essentially **stateless** which is opposed to UCON⁴ which needs to be **statefull**. This difference can be associated with what is in the RedHat’s JBOSS Drools engine, where the default kind of session is statefull (where rules are being constantly checked) whereas there is alternatively the possibility of stateless sessions. This immediately tells us that a rule engine like Drools can accommodate easily both XACML and UCON.
 - There is the choice of calling to an external dedicated XACML engine, or using the rules of Drools to implement the policies (alternatively XACML is implemented by RedHat in the enterprise version of Drools. The specific implementation is called [PicketBox XACML](#). Downside of this is that it only supports XACML 2.0.
 - Other **XACML engines** are included in the major rule engines like coming from IBM or Oracle (but many do not support yet the 3.0 version), but there are very good open source versions as well (see [here](#)), where
 - [Balana](#) is one (which now has become part of the [WSO2 Identity Server](#)) and which supports XACML 3.0
 - A very good resource with examples for Authorization policies is [xacmlinfo.org](#)
 - There exists a graphical editor for XACML from Axiomatics called [ALFA](#) which is a plugin for Eclipse. See [tutorial](#).

2 Documentation: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>

3 https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

4 See survey of Merinelli from 2010 and the paper of Sandhu and Park from 2004

3.2 What can we do with XACML

- There are situations when we need to **combine several policies** into one before making the decision.
“For instance, in a personal privacy application, the owner of the personal information may define certain aspects of disclosure policy, whereas the enterprise that is the custodian of the information may define certain other aspects. In order to render an authorization decision, it must be possible to combine the two separate policies to form the single policy applicable to the request.”
- **Multiple subjects:** is the capability of XACML to specify that for some access decision there are multiple subjects involved, with varying attributes and roles.
A classic example for us is when the Doctor needs to be present (logged in) in order for the Nurse to perform the signature. This is the collaborative access control needed in eHospitals.
- Can do **audit** by using the obligations in which the PDP instructs the PEP to first perform some audit/logging actions before granting access or denying access. Audit is important for eHealth operation environments and process; much work of Hospital IT is concerned with correct auditing... for all health-care processes (like administering medicines, preparing operation rooms, doctor visits).
- XACML is made to be used for describing access control policies and for transmitting them to other components/partner. But the specification always talks about some software components that should handle XACML descriptions, like PDP (decision), PEP (enforcement), PIP (attributes), PAP (policy definitions store). The standard also describes functional requirements for these software components so that any implementation of an XACML engine should conform with these.
The purpose of an XACML policy is just to decide on whether granting access or not, therefore the decision taken by PDP contain either Deny or Permit.
Additionally, a decision can be accompanied by Obligations or Advices. With these the PEP can do more than just deny or permit, i.e. can do auditing.

3.3 Notes about writing XACML policies

For all the below descriptions see the [documentation](#) because here are just simple and extra explanations of the complete and heavy documentation. Simplification also means omitting details, and maybe one is interested in exactly those details. This short descriptions helps with understanding the documentation.

- Each policy (or policy set, which contains several policies or other policy sets) should have a target element specified. The target element is, in logical terms, a **conjunction of disjunctions of conjunctions of Matches** corresponding to respectively a **set of AnyOf of Allof of Match** tags
An empty or missing Target means the policy applies to any request.
Otherwise the policy applies only to those requests for which at least one of each AnyOf tags matches; which for Allof means all Match tags must match.
- All that an operator needs to work with when describing policies are **attributes**, which can be obvious for some, since XACML 3.0 is supposed to specify Attribute based access control policies. But keeping in mind that **everything is an attribute** implies that it works different than in Drools or rule engines, where we match Fact types, which are POJP classes with attributes and accessor functions. The behaviour of Drools in XACML would thus need

a specific attribute that returns the name of the class (which can be done programatically anyway with `instance.getClass().getName()`).

3.4 XACML policies imlementations

- We need a XACML parser, like Balana or PicketBox from JBoss.
- We need an XACML engine, so we can choose PicketBox. But this is not easy to integrate in the Open-source setting (I guess the enterprise one is easy, since a lot of info about the enterprise version exists).
- One could use a XAML translator into Rules.
This is the subject of a MSc topic.
- We can see policies being kept in the same database where the rules are kept, or other Drools resources, like processes.
A client of Tellu can then pick a policy (the same as she picks a rule (template)) and instantiate it to their setting (or apply it), after some possible configuration.

4. Research relevant for ABAC

4.1 Logic based approaches relevant to ABAC

Work on logic based approaches to access control have started more than fifteen years ago with the work of Martin Abadi.⁵ Other approaches involve Epistemic Logics (or logics of knowledge).⁶

4.2 Sticky policies

An alternative to ABAC, or maybe an enhancement, can be the recent concept of Sticky Policies.

4.3 Attribute Based Encryption (ABE)

For general motivation and presentation of ABE and some standard applications see the following papers.

1. Attribute-Based Encryption for Circuits (in JACM)
<http://dl.acm.org/citation.cfm?id=2824233>
2. Secure attribute-based systems
<http://content.iospress.com/articles/journal-of-computer-security/jcs383>
3. Attribute-based encryption for fine-grained access control of encrypted data
<http://dl.acm.org/citation.cfm?id=1180418>
4. Ciphertext-Policy Attribute-Based Encryption
<http://ieeexplore.ieee.org/abstract/document/4223236/>
5. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably

5 M. Abadi, "Logic in access control," *18th Annual IEEE Symposium of Logic in Computer Science, 2003. Proceedings.*, 2003, pp. 228-233. (doi: 10.1109/LICS.2003.1210062)
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1210062&isnumber=27231>

6 Dechesne, F. & Wang, Y. "To know or not to know: epistemic approaches to security protocol verification," *Synthese* (2010) 177(Suppl 1): 51. <https://doi.org/10.1007/s11229-010-9765-8>

Secure Realization

https://link.springer.com/chapter/10.1007/978-3-642-19379-8_4

(giving a reference implementation; see which systems support this)

The topic of ABE applied to Health is very hot now; see the following papers.

1. Flexible Attribute-Based Encryption Applicable to Secure E-Healthcare Records
<https://arxiv.org/pdf/1512.06578.pdf>
2. Self-Protecting Electronic Medical Records Using Attribute-Based Encryption
<https://eprint.iacr.org/2010/565.pdf>
3. Patient-Controlled Attribute-Based Encryption for Secure Electronic Health Records System
<https://link.springer.com/article/10.1007%2Fs10916-016-0621-3>
4. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption
<http://ieeexplore.ieee.org/abstract/document/6171175/>
5. Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation
<https://link.springer.com/article/10.1007/s10207-014-0270-9>
6. Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings
https://link.springer.com/chapter/10.1007/978-3-642-16161-2_6
7. A Review on the State-of-the-Art Privacy-Preserving Approaches in the e-Health Clouds
<http://ieeexplore.ieee.org/abstract/document/6714376/>
8. Secure and Scalable Cloud-Based Architecture for e-Health Wireless Sensor Networks
<http://ieeexplore.ieee.org/abstract/document/6289252/>
(to be checked closer)

4.4 Dynamic ABAC or Usage Control (UCON)