Extending Attribute Based Access Control to Facilitate Trust in eHealth and Other Applications

Jim Longstaff(✉)

Teesside University, Middlesbrough, England
j.j.longstaff@tees.ac.uk

Abstract. We describe a new model for Attribute Based Access Control (ABAC) which handles negative permissions and overrides in a single permissions processing mechanism. The model lends itself to the generation of explanations and permissions review, which can be used to foster end-user trust and confidence in the authorization system. We illustrate using a scenario in which a patient, with the assistance of an information specialist, develops consent directives for her medical records while receiving explanations and demonstrations. The model extends the approaches of ABAC and parameterized Role Based Access Control (RBAC) in that users, operations, and protected objects have properties, which we call classifiers. The simplest form of classifier is an attribute, as defined for ABAC; additional information is also handled by classifiers. Classifier values themselves are hierarchically-structured. A permission consists of a set of classifier values, and permissions review/determining an individual's risk exposure is carried out by database querying. The model has general applicability to areas where tightly-controlled sharing of data and applications, with well-defined overrides, is required.

Keywords: Attribute Based Access Control · Identity and access management · Enterprise information systems

1 Introduction

Trust is recognized as one of the essential factors in the use of computer systems. In authorization, if a user wishes to specify fine-grained controls over access to their data, trust can be gained through explaining the concepts based on a simple underlying model, and providing demonstrations of the access controls.

Attribute Based Access Control (ABAC) is generally considered as the way forward in authorization model research [17]. This reference suggests that implementations and applications will continue apace and also that there is a lack of a formal specifications base for ABAC. A comprehensive description of ABAC is provided in [9]. The central idea of ABAC asserts that access can be determined based on various attribute values presented by a subject. Permissions (often called rules) specify conditions under which access is granted or denied.

M. Felici (Ed.): CSP EU FORUM 2013, CCIS 182, pp. 127–137, 2013.
DOI: 10.1007/978-3-642-41205-9_11,

In this paper we present a new model of permissions which we call the Tees Confidentiality Model, version 2 (TCM2). It is based on a data model suggested by the ANSI standard for Role Based Access Control (RBAC) [1]. The basic RBAC model has the concepts of users being associated with roles, and roles being associated with permissions. Permissions are concerned with performing operations on protected objects. The TCM2 model has the same concepts of users, operations and protected objects; however these concepts now have properties, which we call classifiers, which are used for authorization. The simplest form of classifier corresponds to an attribute, as used in Attribute Based Access Control (ABAC) [9, 12]. User classifiers can take the role of parameters in parameterized RBAC; extended classifiers are defined for combinations of user, operation and protected object, and collection classifiers can be created to facilitate authorizations for collections of objects.

We have a totally different concept of permission to that of RBAC. Our permissions, which we call confidentiality permissions (CPs), in order to distinguish them from RBAC permissions, consist of sets of classifier values. The simplicity of the confidentiality permission format supports very-detailed authorizations and overrides.

TCM2 provides the capability of generating highly-targeted messages to specific users when they would expect to access data which has been denied. The messages could advise on using an override, or could give other information as appropriate.

The paper is organized as follows. Section 2 gives a description of the central concepts of TCM2 which includes examples of permissions and transactions in a notation appropriate for presentation. A comprehensive EHR scenario is presented in Sect. 3. An example of how permissions are processed is given in Sect. 4, while Sect. 5 compares our TCM2 model with current developments in ABAC and RBAC. Conclusions and references follow.

2 TCM2 Overview

2.1 Classifiers

Users, operations and protected objects are described by classifiers. Examples of user classifiers are User_id, UserName, UserRole, Team; for operations Op_id; for protected objects PO_id, PO_Type.

A classifier ordering is determined by the analyst, to indicate importance for matching classifiers. For example if User_id was deemed to be more important to UserRole when determining authorization, then a permission with a User_id value match would be preferred to another permission (not containing a User_id value) which matched by a UserRole value.

There is a type of override operation which allows a user to acquire a more specialized classifier value (if he was authorized to use this override).

2.2 Confidentiality Permissions

A CP is a set of classifier values, which can be provided by several mechanisms (stored database values, generator programs, external applications). Note that CPs

now encompass users, operations and protected objects, in contrast to RBAC permissions.

We now illustrate how CPs are used for authorization. The following CP represents the granting of read-and-append access to psychosis data for a clinician-user in the role "Psychiatrist", under normal (abbreviated "N") processing where no override has been used.

CP1 **Permit_CP (N)**:
{<UserRole, Psychiatrist>,
<Op_id, R_A>,
<PO_Type, Psychiatry>}

Other CPs may be derived using the classifier value hierarchies for every classifier present in a CP. For CP1 above, one derived CP includes the classifier value <UserRole, SeniorPsychiatrist>. Ranges of classifier values can be specified in CPs, as is illustrated later in CP11 – CP13.

A CP will match (i.e. qualify to authorize a transaction) if all its classifier values (describing the user, operation and object) are present in the transaction. Additionally, a CP will match if one of its derived CPs matches. For example the transaction described by

Active Classifier Values:
{<User_id, Fred>,
<UserRole, SeniorPsychiatrist>,
<Op_id, R_A>,
<PO_id, Alice_PsychiatryData >,
<PO_Type, Psychiatry>}

would be matched (and permitted) by permission CP1 above.

2.3 Deny Permissions, Override, and Deny Levels

Deny CPs are negative permissions which prevent access. These can be very detailed, for specific users and data, e.g.

CP2 **Deny_CP (L1)**:
{<User_id, Fred>,
<UserRole, Psychiatrist>,
<Op_id, R_A>,
<PO_id, Alice_PsychiatryData >}

CP2 denies (at Level 1 – see below) Fred access to Alice's psychiatry data when acting in the roles of Psychiatrist (and Senior Psychiatrist). However if authorized to override by the following CP

CP3 **Permit_CP (L1_Ovr)**:
{<User_id, Fred>
<UserRole, SeniorPsychiatrist >,
<Op_id, R_A>,
<PO_id, Alice_PsychiatryData >}

he could use this ‘CP Override’ to cancel the effect of the deny permission CP2, but only if he is acting in the role SeniorPsychiatrist.

Deny CPs are specified at increasing levels of power, called Deny Levels. A deny level contains deny permissions specified at lower deny levels. Therefore data could be denied to certain users who might be able to access it by Level 1 CP Override (if so authorized), whereas more sensitive data might be only available to more senior users who were authorized to override at Level 2.

Deny Levels can be used to implement a Break Glass emergency access approach [5]. The Break Glass approach to authorization has levels of access (pre-staged accounts, emergency levels) providing extra functionality which can be accessed in emergencies. In TCM2, emergency operations and data can be denied at deny levels 1, 2, 3, … etc., where each deny level represents an emergency level which can be accessed using CP Override.

2.4 CP Sets

CPs can be defined as having membership in separate, independent *Confidentiality Permission Sets*, or *CP Sets*. CP Sets can be used separately to determine authorisation, or combined. If a CP Set were to be used in conjunction with other CP Sets, they would first be analysed together for potential conflicts, which would be resolved according to analyst directives. They can be used for several purposes.

CP Sets can represent more detailed Break Glass emergency access than CP override by itself. Here CP Sets will represent access levels containing separately-designed permission sets which can be activated by a break glass override.

Representation of different levels of processing can be accomplished with CP Sets, e.g. Government and State regulations (CPS1), Consumer-specified directives (CPS2), and directives specified by Proxies for Consumers (CPS3). Therefore CPS1 authorizations can be preferred to CPS2 authorisations, if this is what the application requires.

In the examples that follow, normal access to health records (Sect. 3.2) could be provided by one CP Set, and the restrictions placed by an individual user on their own health records (Sects. 3.3−3.5) could form another CP Set.

3 HealthCare Scenario

3.1 Overview

We now give a comprehensive example of CPs, and briefly illustrate their processing in Sect. 4. The example concerns Electronic Health Record (EHR) data for a single patient Alice, and the consent directives represented by CPs which permit and restrict access to it. The patient develops the CPs through interacting with an information specialist.

3.2 Normal Access to EHR Data by Healthcare Professionals

For convenience we refer to this as access to Unrestricted Data (UD) for healthcare professionals. EHR data is to be made available to

(a) Healthcare professionals (HCPs) such as clinicians, doctors, and administrators who have a Legitimate Relationship or LR with the patient (specified by CP4).
(b) Additionally, all HCPs can exercise a Level 1 CP Override facility (CP5).

CP4 **Permit_CP (N)**: {<UserRole, HCP>, <LR, yes>, <Op_id, R_A>, <PO_Type, EHR>}	CP5 **Permit_CP(L1_Ovr)**: {<UserRole, HCP>, <LR, yes>, <Op_id, R_A>, <PO_Type, EHR>}

3.3 Access to Data Restricted by Consent Directive 1 (The Most Sensitive Data)

This data (several documents and database records), referred to as CD1 data, is to be made available to

(a) Certain individuals, Bob and Harry, when they are using role Senior Psychiatrist (CP6 and CP8).
(b) Bill, the author of this data, will also be able to access it, providing he is in a healthcare role, and has a Legitimate Relationship with the patient (CP6 and CP9).
(c) Other users on role Senior Psychiatrist, without an LR, on Level 2 CP Override, which is a privileged facility not generally available to healthcare professionals. (CP6 and CP10). (A message to Senior Psychiatrists would be displayed, indicating they could access restricted data by Level 2 CP Override.)

A further message is to be generated for the patient's GP, Fred, when he is denied access to this data through the matching of CP7. This message might say that the patient has restricted vital data from him, and suggest that he contacts a person who could access this data.

CP6 **Deny CP(L2)**: {<UserRole, HCP>, <PO_Coll_id,Alice_poc1>, <PO_Type, EHR>}	CP7 **Deny_CP(L2)**: { <User_id, Fred>, <UserRole, GP>, <PO_Coll_id, Alice_poc1>, <PO_Type, EHR>}	CP8 **Permit_CP (N)**: {<User_id, < Bob, Harry> >, <UserRole,SeniorPsychiatrist>, <Op_id, R_A>, <PO_Coll_id, Alice_poc1>, <PO_Type, EHR> }
CP9 **Permit_CP (N)**: {<User_id, Bill>, <UserRole, HCP>, <LR, yes>, <Op_id, R_A>, <PO_Coll_id, Alice_poc1>, <PO_Type, EHR>}	CP10**Permit_CP(L2_Ovr):** {<UserRole, SeniorPsychiatrist >, <Op_id, R_A>, <PO_Coll_id, Alice_poc1 >, <PO_Type, EHR>}	

3.4 Access to Data Restricted by Consent Directive 2

This (CD2) data (database records written by a particular clinician over a certain time period at a specific site) is to be made available to users as follows:

(a) All HCP users on Level 2 Override (if they are authorized to use it for this data).
(b) Senior Psychiatrist users on Level 1 Override, but only providing a precondition PC1 is true.

General messages indicating the availability of restricted data following override would be displayed. If the CP Level 2 Override were to be used, this would include a CP Level 1 Override, irrespective of the precondition evaluation).

CP11 **Deny_CP (L2)**: {<UserRole, HCP>, <Op_id, R_A>, <PO_id, Alice_EHR>, <PO_Type, EHR>, <PO_clinician_of_care, Bill>, <PO_EndDate, 01jan03>, <PO_Site, 2>}	CP12 **Deny_CP (L1)**: {<UserRole,SeniorPsychiatrist>, <Op_id, R_A>, <PO_id, Alice_EHR>, <PO_Type, EHR>, <PO_clinician_of_care, Bill>, <PO_EndDate, 01jan03>, <PO_Site, 2>}	CP13 **Permit_CP (L1_Ovr)**: {<UserRole, SeniorPsychiatrist>, <Op_id, R_A>, <PO_id, Alice_EHR>, <PO_Type, EHR>, <PO_clinician_of_care, Bill>, <PO_EndDate, 01jan03>, <PO_Site, 2>, <PC1, true>}
CP14 **Permit_CP (L2_Ovr)**: {<UserRole, HCP>, <Op_id, R_A> <PO_id, Alice_EHR>, <PO_Type, EHR>, <PO_clinician_of_care, Bill>, <PO_EndDate, 01jan03>, <PO_Site, 2>}		

3.5 Access to Data Restricted by Consent Directive 3

This (CD3) data (all psychiatric data) is to be made available to

(a) Senior Psychiatrists.
(b) Other HCP users via Level 1 CP Override.

CP15 **Deny_CP(L1)**: {<User_Role, HCP>, <Op_id, R_A>, <PO_id, Alice_EHR>, <PO_Type, Psychosis>}	CP16 **Permit_CP(N) :** {<UserRole,SeniorPsychiatrist>, <Op_id, R_A>, <PO_id, Alice_EHR>, <PO_Type, Psychosis>}	CP17 **Deny_CP (L1)**: {<UserRole,SeniorPsychiatrist>, <Op_id, R_A>, <PO_id, Alice_EHR>, <PO_Type, Psychosis>, <PO_clinician_of_care, Bill>, <PO_EndDate, 01jan03>, <PO_Site, 2>}

An additional message to Senior Psychiatrist users following the display of psychiatric data might alert to the availability of further (restricted) data (Consent Directives 1 and 2) through override. This would be to guard against possible oversight by the user, which might have serious consequences for the treatment of the patient.

3.6 An Authorization Example

As a consequence of the consent directives, consider an authenticated user (John) with an activated role of Senior Psychiatrist who queries the EHR for patient Alice, with whom he has an LR. For Normal Processing and Level 1 Override Processing, access to UD and CD3 data is permitted; for Level 2 Override Processing, UD, CD3, CD2, and CD1 objects are permitted.

4 CP Processing

Space restrictions only permit a brief illustration of how permissions are processed. CP processing depends on two principles. Firstly, a CP will match (i.e. qualify to authorize a transaction) if all its classifier values are contained in the transaction. Additionally, a CP will match if one of its derived CPs matches (An example of this was previously provided in Sect. 2.2).

The second principle concerns determining which of two CPs (taken from a set of Matched CPs) is the stronger or nearer match to a transaction. This Nearest Match CP would then have a higher priority in determining the authorization outcome. An elaboration of this principle follows.

A confidentiality permission is a set of classifier values. There is an ordering *cfiersq* on the classifiers that is set by the security architect and is a mapping of the set of integers 1, 2, 3, 4 … to the set of classifiers.

In order to compare two CPs *cp1*, *cp2* to find which one is the nearer match, we first determine the lowest classifier ordering number for each CP, namely $NCFIER_L(cp1)$ and $NCFIER_L(cp2)$.

If $NCFIER_L(cp1)$ and $NCFIER_L(cp2)$ are not the same then the CP with the lower classifier ordering number, i.e. the most important classifier specified by the security architect, is the nearer match. If they are the same then we consider the classifier values for these lowest ordering number classifiers, namely $VCFIER_L$ (*cp1*) and $VCFIER_L$ (*cp2*).

If $VCFIER_L$ (*cp1*) and $VCFIER_L$ (*cp2*) are not the same then we determine if they are related through the ancestor/descendant relationship and if so the descendant value takes priority and determines the Nearer Match CP.

If they do not have an ancestor/descendant association, or they are the same, then the whole process is repeated with the next lowest ordering number classifier for each CP, and so on, until the nearer match is obtained.

These principles can be used to directly determine the authorization outcome for a user, operation or protected object. Alternatively, by partially matching on user classifier values, they can be used to generate a sequence of Nearest Matched CPs which can be processed by database querying; this is illustrated in the examples below.

We now describe the processing of the transaction given in Sect. 3.6. The transaction is restricted to a single EHR for a single patient (Alice), and its sub objects. All Permit and Deny CPs specified the R_A operation. The initial set of Matched CPs, and the Nearest Match CP sequence (in order of strength of matching, and following Override CP removal for normal processing) are

Initially Matched CPs		**Nearest Matched CPs (no overrides)**		
CP4	Permit_CP (N)	CP4	Permit_CP (N)	1
CP5	Permit_CP (L1_Ovr)	CP6	Deny_CP (L2)	7
CP6	Deny_CP (L2)	CP11	Deny_CP (L2)	2
CP10	Permit_CP (L2_Ovr)	CP12	Deny_CP (L1)	4
CP11	Deny_CP (L2)	CP15	Deny_CP (L1)	3
CP12	Deny_CP (L1)	CP16	Permit_CP (N)	5
CP14	Permit_CP (L2_Ovr)	CP17	Deny_CP (L1)	6
CP15	Deny_CP (L1)			
CP16	Permit_CP (N)			
CP17	Deny_CP (L1)			

The match strength is indicated in ascending order, starting with the weakest (i.e. 1). Processing this Nearest Match CP sequence authorizes the retrieval of UD and CD3 objects. During demonstrations of the effects of permissions to the patient, it can be pointed which CPs permit or deny access to data, e.g. a particular CD1 object is denied by CP6, while a CD3 object is permitted by CP16.

Now suppose overrides are used. The same initially-matched CPs are returned. However on applying CP Override at Level 1 (involving deletion of L2_Ovr CPs), removing delete-related CPs and applying Nearest Match, the sequence of Nearest Matched CPs shown below is obtained: processing this sequence also determines that access is permitted to UD and CD3 data. If L2_Ovr is used, the indicated sequence is obtained: these CPs authorize access to UD, CD1, CD2, and CD3 data.

Nearest Match CPs (L1_ovr)			**Nearest Match CPs (L2_ovr)**		
CP4	Permit_CP (N)	1	CP4	Permit_CP (N)	1
CP5	Permit_CP (L1_Ovr)	2	CP5	Permit_CP (L1_Ovr)	2
CP6	Deny_CP (L2)	7	CP6	Deny_CP (L2)	5
CP11	Deny_CP (L2)	3	CP10	Permit_CP (L2_Ovr)	6
CP12	Deny_CP (L1)	4	CP14	Permit_CP (L2_Ovr)	3
CP16	Permit_CP (N)	5	CP16	Permit_CP (N)	4
CP17	Deny_CP (L1)	6			

5 Related Work

TCM2 has a number of similarities with our previously published TCM work [14, 15]. Role is treated as an application concept, and similar overrides are proposed. Also, the previous TCM papers have described design and processing strategies for permission types, but not for dynamic authorization involving individual permissions, as has been presented in this paper.

In the original TCM, hierarchies of classifier collections formed the basis of permissions processing, and permissions design. Also inheritance of permissions within classifier collection hierarchies was specified using permission types. In this paper, hierarchies of classifier values, with permissions inheritance always assumed, replaces classifier collection hierarchies. This is a major difference and simplification between TCM2 and the TCM. Also the TCM has no concept of permission-triggered messages.

Regarding emergency access to data, the Break-Glass approach [5] provides emergency accounts giving access to normally restricted data. The difficulties of such an approach are discussed in [6], which integrates a break glass approach into access control software; emergency level access is supported. These emergency levels are similar to the 'deny levels' concept in TCM2.

A large research development in RBAC can be described under the term parameterized RBAC. Part of this work involves using external (sometimes called contextual, environmental) information to control the processing of roles, and therefore provides additional functionality over standard RBAC, including what could be described as an override capability [2, 19]. The TCM2 model provides aspects of external parameter handling as part of its basic model and design framework (in that classifiers can represent external parameters). This approach is similar to that advocated in [7], which prefers the use of 'role attributes' to the use of external policy-enforcing systems (where this is possible).

Recently, investigations into Attribute Based Access Control, or ABAC, have been carried out, to address the inflexibility to change of RBAC models, and authorization models to be used for distributed applications [4, 8, 9, 12]. Access decisions are based on attributes that the user can be proved to have. In ABAC, different parties must reach trust agreements over attribute definitions, which can be more straightforward than agreeing consistent role definitions. ABAC provides good support for context, such as time of day. ABAC has been sometimes referred to as Policy Based Access Control or Claims Based Access Control. ABAC research, particularly focusing on attribute integrity and security, has been referred to as the 'grand challenge', and the future direction of authorization model development [17]. Applications in messaging and cryptography are described in [13, 21], and consistency and fault detection in rule structures is reported in [11].

XACML is an extensively developed and implemented ABAC approach, for which the underlying model has similarities with TCM2. XACML subjects, actions, and resources (corresponding to TCM2 users, operations, and protected objects) have attributes, on which authorization decisions are made. A comprehensive architecture involving PDPs, PEPs is defined for this. There is provision for extensions to be

written into an XACML application, which could be used in an implementation of TCM2.

TCM2 has additional concepts, namely Levels of Access, and overrides. A significant difference between TMC2 and XACML appears to be in simplicity of use. TCM2 very simply facilitates the modeling and use of authorization concepts, such as hierarchically-structured attribute values, inheritance of permissions. An industrial strength TMC2 implementation would use full relational database, with efficient management of large volumes of data and permissions, and direct database programming of permissions processing, which is essentially simple. An efficiency study for large XACML applications is reported in [16].

There are potential difficulties for permissions review/risk exposure for ABAC—potentially large numbers of rules, and their processing, must be considered. Reference [10] proposes a combination of ABAC and RBAC, in which the permissions available to a user are the intersection of permissions provided by RBAC active roles and ABAC rules. The TCM2 model extends the ABAC approach in that classifiers (which can represent ABAC attributes) are defined for operations and protected objects, in addition to users. Note that there is no direct TCM2 equivalent to RBAC permissions, which are used in the presentation of ABAC models. A recent approach to combining RBAC and ABAC was presented in [14]; TCM2 permissions can be designed to directly implement RBAC authorization.

6 Conclusions

We have described an ABAC permissions structure which directly supports fine-grained authorization, overrides, and highly-targeted messages produced during permissions execution. A formal specification of our model has been developed using the B-Method [3, 18], which is too detailed to be covered in this paper. As far as we are aware, no other model provides all this functionality in a simple and straightforward way. The functionality enables the support of the interactive development of permissions, during which permissions review ("who can see what") and deny/override schemes can be explored and demonstrated. It has been our experience that this feedback engenders trust in the authorization system on the part of non-technical end users.

We have provided a detailed healthcare records authorization scenario which demonstrates the modeling power of our approach, and we have briefly described the feedback and messages provided to the end user to promote trust. Throughout the development of the TCM and TCM2 models, close collaboration with the England healthcare authorities (the NHS) has taken place, and the ideas have contributed to the current authorization scheme mandated by the NHS [20], and to its previous versions.

Acknowledgment. The author wishes to thank Tony Howitt, Professor Mike Lockyer, Professor Michael Thick and Steve Dunne for advice and contributions. The work was supported in part by grants and contracts from the England NHS National Programme for IT, particularly as part of the ERDIP and HRI Programmes (2000−2006).

References

1. ANSI 2012, American National Standard for Information Technology: Role Based Access Control, ANSI INCITS 359-2012. www.incits.org (2012)
2. Bacon, J., Moody, K., Yao, W.: A model of OASIS role-based access control and its support for active security. ACM Trans. Inf. Syst. Secur. **5**(4), 492–540 (2002)
3. B-Method: www.methode-b.com (2013)
4. Blaze, M., Feigenbaum, J., Ioannidis, J.: The KeyNote Trust Management System Version 2. IETF RFC 2704. http://www1.cs.columbia.edu/~angelos/Papers/rfc2704.txt (1999)
5. BREAK-GLASS (SPC): Break-glass: an approach to granting emergency access to healthcare systems. White paper, joint NEMA/COCIR/JIRA Security and Privacy Committee (2004)
6. Brucker, A.D., Petritsch, H.: Extending access control models with break-glass. In: Proceedings of 2009 ACM Symposium on Access Control Models and Technologies (2009)
7. Goh, C., Baldwin, A.: Towards a more complete model of role. In: Proceedings of Third ACM Workshop on Role-Based Access Control (1998)
8. Karp, A.H., Haury, H., Davis, M.H.: From ABAC to ZBAC: the evolution of access control models. Tech. Report HPL-2009-30, HP Labs (2009)
9. Hu, V.C., Ferraiolo, D., Kuhn, R., et al.: Guide to Attribute based Access Control (ABAC) Definition and Considerations (Draft). NIST Spec. Publ. 800-162. http://csrc.nist.gov/publications/drafts/800-162/sp800_162_draft.pdf (2013)
10. Huang, J., Nicol, D.M., Bobba, R., Huh, J.H.: A framework integrating attribute-based policies into role-based access control. In: SACMAT '12, Newark, New Jersey, USA (2012)
11. Kuhn, D.R.: Vulnerability hierarchies in access control configurations. In: 4th Symposium on Configuration Analytics and Automation. IEEE (2011)
12. Kuhn, D.R., Coyne, E.J., Weil, T.R.: Adding attributes to role-based access control. IEEE Comput. **43**(6), 79–81 (2010)
13. Li, J., et al.: Attribute-based signature and its applications. In: ASIACCS '10, Beijing, China, 13–16 April (2010)
14. Longstaff, J.J., Lockyer, M.A., Nicholas, J.: The tees confidentiality model: an authorization model for identities and roles. In: Proceedings of Eighth ACM Symposium on Access Control Models and Technologies (2003)
15. Longstaff, J.J., Lockyer, M.A., Howitt, A.: Functionality and implementation issues for complex authorization models. IEE Proc. Softw. Special Issue (on Role Based Access Control) **153**(1), 7−15 (2006) ISSN 1462-5970
16. Ros, S.P., Lischka, M., Marmol, F.G.: Graph-based XACML evaluation. In: SACMAT '12, Newark, New Jersey, 20–22 June (2012)
17. Sandhu, R.: The authorization leap from rights to attributes: maturation or chaos? In: SACMAT '12, Newark, New Jersey (2012)
18. Schneider, S.: The B-Method: An Introduction. Palgrave, Basingstoke (2001)
19. Stermbeck, M., Neuman, G.: An integrated approach to engineer and enforce context constraints in RBAC environments. ACM Trans. Inf. Syst. Secur. **7**(3), 392–427 (2004)
20. UK NHS: Care Records Guarantee. http://www.nigb.nhs.uk/pubs/nhscrg.pdf (2011)
21. Yu, S., et al.: Attribute based data sharing with attribute revocation. In: ASIACCS'10, Beijing, China, 13–16 April (2010)