Personalized Access Control for a Personally Controlled Health Record

Lillian Røstad
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
lilliaro@idi.ntnu.no

Øystein Nytrø
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
nytroe@idi.ntnu.no

ABSTRACT

Access control is a key feature of healthcare systems. Up until recently most healthcare information systems have been local to a healthcare facility and accessible only to clinicians. Currently there is a move towards making health information more accessible to patients. One example is the Personally Controlled Health Record (PCHR) where the patient is in charge of deciding who gets access to the information. In the PCHR the patient is the administrator of access control. While it certainly is possible to create roles representing people most patients would want to share with, like primary physician, it is also likely, and desirable, to afford the patients a high level of control and freedom to be able to create specialized access policies tailored to their personal wishes. We entitle this *personalized access control*. In this paper we present a semi-formal model for how we believe personalized access control may be realized. The model draws on and combines properties and concepts of both Role-Based Access Control (RBAC) and Discretionary Access Control (DAC) to achieve the desired properties. Throughout the paper we use the PCHR as a motivating example and to explain our reasoning and practical use of the model.

Categories and Subject Descriptors

H.1 [**Information Systems Models and Principles**]: Miscellaneous

General Terms

Security

Keywords

Access Control

CSAW'08, October 31, 2008, Fairfax, Virginia, USA.
Copyright 2008 ACM 978-1-60558-300-6/08/10 ...$5.00.

1. INTRODUCTION

Access control is a key feature of healthcare systems. Enforcing access control on sensitive health data is about protecting the patient's privacy as well as ensuring that clinical personnel have access to the information they need to provide the best possible care. Access control has a unique challenge in that it is always most important to save the patient's life. In other words: though confidentiality is the norm - availability takes precedence when the patient's health is at stake.

A challenge in healthcare today is the lack of connectivity and sharing. Information exists in proprietary information systems local to hospitals or doctors offices and accessible only to health care personnel. Personal Health Records (PHR) have been proposed as a potential solution to this problem. The term PHR has been defined by The Markle Foundation as:

> ”An electronic application through which individuals can access, manage and share their health information, and that of others for whom they are authorized, in a private, secure, and confidential environment.”[1]

The challenge with most PHRs is that they are local and specific to one point of care [4] and therefore most existing PHRs only contain a subset of a patient's clinical information. The Personally Controlled Health Record (PCHR) has been proposed as a possibility that has the potential to solve many of the PHR's shortcomings. The goal of a *Patient/Personally Controlled* Health Record (PCHR) [5] is to assemble the patient's complete health history by importing data from many source systems. A PCHR differs from a PHR in that it exists outside of organizational boundaries and contains data from multiple care sites. Also, the patient is in complete control of the information in the PCHR. The patient decides what data should be added to the PCHR. Any data import has to be approved by the patient. The patient also decides who gets access to the information in the PCHR. This means that it is the patient who is administrator of access control [7]. Through a PCHR the patient may choose to share his data with health care providers, family members and any other as needed.

One of the main challenges of the PCHR is the duality of empowerment potential and privacy risk. The patient is empowered in that he is given control over his own health information. But, it may also increase the risk of inadvertently leaking sensitive information about himself as the patient is solely in charge of assigning access rights and maintaining these over time. This means that it is important to have an access control model that is easy to use, hard to misuse, yet affords the patient a high-level of flexibility and control.

In many healthcare systems today, Role-Based Access Control (RBAC) is the norm. Healthcare organizations fits very well the RBAC premises of having many users that can be grouped into a relatively small number of roles. One may argue that for a PCHR this is also true. For instance, most people would likely want to give their primary physician access to their PCHR. However, it is also likely that given the opportunity to share with anyone, many users will construct access policies that are personal, unique and not generalizable. As such there is a need for a model that has both a pre-defined, common set of access policies, for convenience, yet allows the patient absolute control when desired.

In this paper we present a semi-formal model for what we have entitled *personalized access control*. The model is motivated by our work on personal health records [7] and the PCHR is used as a motivating example throughout the paper. The model is semi-formal in the sense that some properties still requires some more discussion and there remains some issues to be resolved.

2. THE PERSONALLY CONTROLLED HEALTH RECORD

A PCHR is a collection of clinical information about a patient [5]. What's unique about the PCHR is that the patient is in charge of deciding who gets access to this information by assigning sharing privileges. In this section we provide some usage scenarios to help explain the PCHR concept in more detail and how it will be used. These examples will be used for explanation throughout this paper as we move from requirements to a more detailed description of personalized access control in a PCHR.

PCHR usage scenarios

1. A patient moves from one city to another. She decides to give her new primary physician access to her PCHR so he can read up on her medical history before their first appointment. She also decides that he should be able to add information to her PCHR, so she will have a complete medical history there in case she has to move again.

2. A patient that has been healthy most of his life, suddenly is diagnosed with a complex disorder. This diagnosis implies that he will from now on need regular services from many health care providers including a physical therapist, an orthopaedist and an occupational therapist in addition to his primary physician. To provide the best care it is helpful if all the service providers are aware of and informed about the other services he receives and how they are progressing. The patient decides to set up a PCHR and grant all of his providers access to read the information in his PCHR.

3. A young girl has had a PCHR for a while. The girl is now 17 and still not legally an adult, but as she is considered an adolescent, she is in control of the PCHR and her parents currently do not have any access. One day she has an accident on the way to school an breaks her leg. The X-ray summaries and the doctor's notes are added to the PCHR as is routine. Her mother is concerned and asks if she can get access to the PCHR so she can read the information. Using the PCHR it is possible for the girl to give her mother access only to the parts of the PCHR that she considers ok to share. The mother is never aware of what she cannot see.

3. REQUIREMENTS FOR PERSONALIZED ACCESS CONTROL

Based on the previous section, we can formalize a set of requirements for our model for personalized access control for a personally controlled health record:

1. The patient is the owner of information in the PCHR.

2. Every information element in the PCHR database is owned by somebody.

3. Any information element in the PCHR database has only one owner.

4. The patient is administrator of access to his/her information. The patient decides what permissions to assign to who.

5. Information in the PCHR is structured in categories. Examples of categories include: lab results, clinical notes, immunizations etc.

6. Every information element in the PCHR belongs to a category.

7. Permissions may be granted on a category or a single information element.

8. Permissions are granted by assigning an access policy to another user. An access policy is a set of permissions.

9. For ease of use it should be possible to define a set of access policies believed to be common to most users (patients).

10. For reuse purposes it should be possible for the patient to create personal access policies.

11. For simplicity it should be possible for the patient to create a new access policy by adapting one of the common policies to his/her specific needs, or by extending or adapting one of his/her personal policies.

12. The patient should not be allowed to update or delete the common access policies.

13. For flexibility the patient should be allowed to update or delete any of his/her self-defined access policies at any time.

14. The patient may at any time revoke an assigned access policy.

4. RELATED WORK

To the best of our knowledge, no model with these exact properties have been proposed before. However, there exists work that has similarities. Most notably there are similarities to both Role-Based Access Control (RBAC)[2] and Discretionary Access Control (DAC). The concept of access policies in our model is similar to roles in RBAC. We will reuse many of the RBAC properties for combining and applying roles to our access policies. We have chosen to use the term *access policy* rather than *role* because in our model an access policy may be personal and specialized while a role in RBAC is supposed to be generalized and "define once - apply many".

In the RBAC standard [2] a role is defined as: *(..) a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role.* In this traditional definition of a role, the role describes a relation between a person/set of persons and a set of objects. In our work, an access policy represents a relationship between two persons - the owner of the information in the PCHR and the person she is sharing her information with. This is significantly different from standard RBAC where a role is a mapping from a user to allowed actions on data.

Our work has similarities with DAC in that information has an owner, and the owner has discretionary authority over who else can access that information. In [6] Osborn et al. presents how DAC may be implemented using RBAC. In their approach they create an owner role that is associated with each object, and declares ownership by assigning this role to a user. We will not adopt this approach since, as already stated, we are not using RBAC directly and also because while that approach shows that it is possible to implement DAC using RBAC it is not uncomplicated. Also, in DAC it is usually the case that the owner of an object is the one who created the object. For our model the owner is the one the information is about. The owner may allow others to add information, but the information created belongs to the owner of the record it is part of.

In our model we need to allow negative permissions. That is, we want to be able to combine access policies to create an adapted policy that contains most of the permissions of the policies it is based on, but with some exceptions. Negative authorizations have been proposed in [3] where attribute expressions are used to prevent a user from being able to assume a role. The issues of potential conflict in negative authorizations are relevant for the use of negative permissions in our model.

5. PERSONALIZED ACCESS CONTROL

In this section we present the core components of our model for personalized access control. From this point on we will use the abbreviation *PAC* for Personalized Access Control.

We start out by defining an access policy as:

Definition 1. An **access policy** is a representation of a relationship between two people. This relationship is reflected in the permissions one user (the owner) grants another user through policy assignment.

Throughout the remainder of this paper we will assume *access policy* and *policy* to have the same meaning. We begin our discussion by elaborating on some of the requirements and from that we construct the core PAC model.

Central to the PAC model is the concept of ownership of information. Every information element in a PCHR is owned by the patient and only the owner may decide who to share information with. In other words only the owner has the power to assign and revoke permissions. And the owner can of course only share her own information. As stated in the requirements an information element has to have one, and only one, owner. Any information created in or added to the PCHR is owned by the patient: ownership is not linked to who creates information, but to who owns the PCHR the information is part of.

A PCHR may over time grow very large. Therefore it does not seem like a good solution to only have the possibility of setting permissions on single information elements. However, we may take advantage of the fact that most healthcare information is heterogeneous, often with complex structure, types and relationships. Information is often grouped by topic - e.g. doctor's notes, immunizations, x-rays etc. The specific information may be complex or simple, we just need a category tag to identify parts of the structure. A category "personal information" may subsume another category "allergies". Note that access to "personal information" and access to "allergies" may conflict, and can only be resolved by taking the information structure into account. The actual meaning of the permissions given by a policy will thus have to be interpreted according to the information model. To simplify the PAC model, for now, we simply state that any information must belong to a category. Note that no restrictions are placed on the number of categories an information element may belong to. This has to be included for practical reasons, though it does lead to some complications when combining and interpreting policies that we will discuss further later on in this paper.

As stated in the requirements, to allow for specific control and high granularity, in the PAC model it is possible to grant permissions on both specific information elements and categories. The assumption is that most of the time granting permissions on categories is sufficiently detailed. Permissions on information elements will probably only be used in specific situations e.g. like in the example of the girl with the broken leg. In general she wants her mother to see her lab results, but not the ones related to the abortion. Also, the existence of categories makes it possible to construct generalized policies, or policy templates, that can be reused.

To fulfill requirements 9-13, we need to construct two sets of access policies in our model. We denote these two policy sets *common policies* and *personal policies.* The common policies are not changeable by the patient while the patient is in complete control of the personalized policies. The set of common policies, describing common relationships, should be system-wide and available to all users. As such there exists only one common policy set while there are just as many personalized policy sets as there are users that are information owners in the system, as depicted in figure 1. Note that the personal policy set for a user may be empty. We will return to the personal and common policies and discuss them in more detail after we have defined the core PAc model.

In the PAC model an access policy, common or personal, may exist without being assigned to anyone. An unassigned

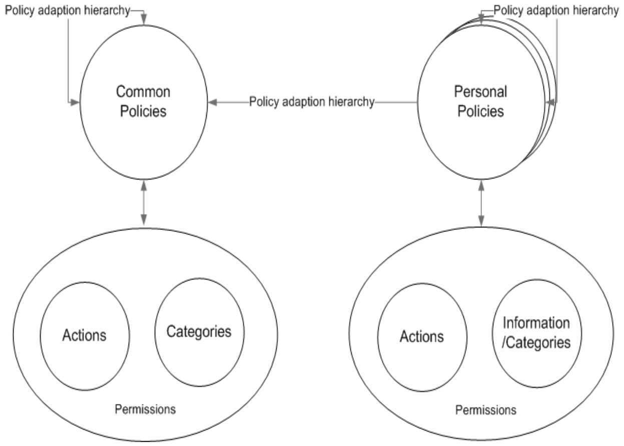

Figure 1: Common and personal policies

policy represents a potential relationship. Assigning the polcy to a user means establishing a relationship between the assigner and the assignee. An example of a potential relationship is that of a *primary physician* as described in scenario 1. This is a relationship that most people have with one person. Most people would probably also agree that your primary physician should have access to most of your clinical information and also be able to add information. It is probably possible to create a common access policy for primary physicians with a minimal set of permissions that most people would agree is representative and appropriate. But it is only when the owner of a PCHR assigns this role to a person that the relationship *user u is primary care provider for owner o* is established.

There are mainly two sets of users: regular users and owners. Not every user has to own data. But only users owning data have a personal policy set and the ability to share their information.

5.1 Core PAC model

With this in mind, we start out by defining the core concepts of the PAC model more formally. Then we move on to discussing policy definition, assignment and revocation in more detail. Figure 2 depicts the Core PAC Model. The figure shows how the relationship between two users is established by policy assignment. It also shows how a permission, in an assigned policy, is a set of allowed actions on information elements and that there is a *owner* relationship linking a user (the owner) directly to information elements. This model is similar to the core RBAC model [2]. The main difference is the introduction of ownership and that a policy links two users. Figure 1 illustrates that the model consists of two policy sets: a set of policies that are common and known to all users and a set of personal policies that are specific to one user.

From this we define the core components of the PAC model:

- A set U of *users*
- A set O of *owners* where $O \subseteq U$
- A set C of *categories*
- A set I of *information elements*

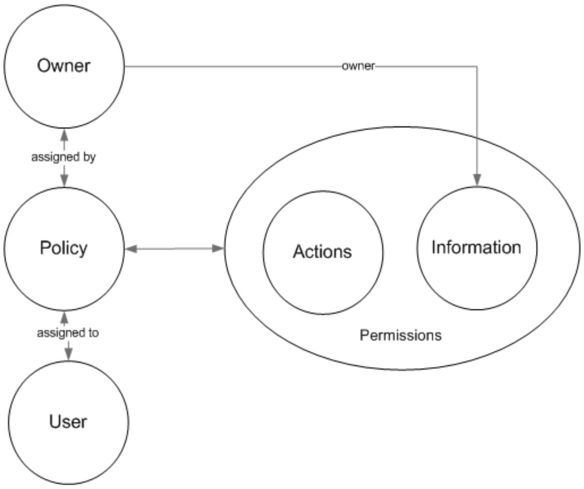

Figure 2: Core PAC Model

- A set A of *actions*
- A set CP of *category permissions* $CP = \{(a, c) | a \in A, c \in C\}$
- A set IP of *information permissions.* Permissions on single information elements. $IP = \{(a, i) | a \in A, i \in I\}$
- A set $CPol$ of *common policies*
- A set $PPol$ of *personal policies*
- A *common policy* is a set of category permissions, i.e. $CPol \subseteq CP$.
- A *personal policy* is a set of category permissions and information permissions, ie. $PPol \subseteq CP \times IP$.
- A function ci from a category to the set of all information elements (possibly empty) belonging to that category.
- A function io from an information element to the (unique) user owning that information.

The concepts of common and personal policies, and hierarchies of such, are described in more detail in the following sections.

5.1.1 Common policies

Common policies form policy hierarchies. These hierarchies differ from hierarchical RBAC in that they are not hierarchies under subsumption of a set of permissions, but policies related by being derivable from each other from root to node according to simple rules of addition and removal of permissions. For example, the policy “Significant other” could be related to the superior policy “Family” by adding access to all information of category “Medication” but explicitly remove (just in case it would be accessible) all information in category “Childhood diagnoses”. Thus the hierarchies form by virtue of the order that permissions are removed and added. We denote this an *adaption hierarchy.*

It is worth noting here what adapting means. In RBAC, when one role extends another we say that there is an inheritance relationship between the roles. The RBAC standard

[2] defines a role inheritance relationship as: *role r_1 **inherits** role r_2 if all privileges of r_2 are also privileges of r_1*. In the case of PAC we want to be able to base one policy on another, but we also want the flexibility to allow this new policy to be both *wider* and *more narrow* than the policy it is based on. In other words we want to be able to both add and subtract privileges, which means that we need to be able to specify negative privileges. Usually, when roles are combined the default rule is *permit overrides*. If one of the roles that are combined allows, then the result is allow. Negative privileges are only represented as the absence of a privilege, and the resulting permission set is the combination of all privileges in the roles that are combined. An actual role is just a set of (positive) category permissions resulting from this process. The permissions of an *adapted policy* is calculated by adding together all the positive permissions, removing a permission if any of the participating policies has a negative permission for this category or information element. As for roles the result of this calculation is a set of only positive permissions, and absence of a permission implies no permission. So the result of policy adaption is the same as when role hierarchies are collapsed, but the calculation process is different. In the calculation process for policy adaption the rule for policy combination is *deny overrides*.

5.1.2 Personal policies

Every user that owns information potentially has a set of personal policies. The set may be empty. As Figure 1 shows, the personal policies differ from the common policies in that they are a mapping of allowed actions on *owned information elements* and/or categories. A personal policy may be more specific and detailed than a common policy. This is necessary to cover those situations where a patient wants to share some of the information belonging to a category, but not all. For instance a patient may want to share knowledge of some of her test results with her mother, but not all of them, as illustrated in example 3.

As for common policies we also use the concept of *policy adaption* for personal policy. A personal policy may:

- Be an independent entity.
- Be based on one or more personal policies by an adaption relation.
- Be based on one or more common policies by an adaption relation.
- Be based on common *and* personal policies by adaption relations.

Figure 3 provides an example adaption policy to illustrate the concept. The top node is the empty set. Note that the set contains two sub-sets: the set of positive permissions and the set of negative permissions. Each policy in the hierarchy consists of a positive and a negative permission set.

5.2 Policy definition

Policy definition is about creating a set of permissions and declaring adaption relationships to other policies. Defining a common policy is simpler than a personal policy because we only deal with categories and there is no concept of ownership. For definition of personal policies we also need to include individual information elements related to an owner. We need to state some rules for adaption relationships on policy definition:

- Adaption relationships form a lattice.
- Every element in the lattice is composed of two sets: a set of positive and a set of negative permissions.
- A permission is an allowed action on a category or an information element.
- The resulting policy is calculated by collapsing the adaption hierarchy in the definition. The last element to be added is the set of specific permissions defined for the policy to be calculated.
- *Deny overrides* is used as the rule for calculation. If any policy denies a permission, then that permission is left out of the resulting calculation.
- The resulting policy is a set of positive permissions.

5.3 Policy assignment

In PAC policy assignment is interpreted as *relationship declaration*. Unassigned common and personal policies may exist. An unassigned policy is simply a policy definition. Figure 2 illustrates relationships in PAC. A relationship is a direct link between two users established through a policy assignment.

An owner assigns a policy (declares a relationship) by:

- Assigning a common policy.
- Assigning a personal policy.

Note that the assigned policy may depend on any number of other policies by definition.

A policy assignment is a one-to-many relationship between an owner and other users. An owner may share her PCHR with many other users. An owner may also assign multiple policies to the same user. This is required to handle situations like when one person is both the father of and e.g. physiotherapist for one patient. In multiple policy assignments the permissions of the policies are combined. For this combination we apply the rule of *permit overrides*, and as such it is different from policy adaption. The resulting permissions are the sum of permissions in all assigned policies. If the policies to be combined are themselves defined in terms of adaption hierarchies, the adaption calculations are performed first and then the resulting policies are combined. Remember that the result of calculating an adaption is a policy containing only positive permissions. We will illustrate this process by example in the next section.

5.4 Policy activation

A policy is activated when a user applies the policy to access information. A policy definition is simply a declaration of how one policy is related to others, and what to add or remove specifically for this policy. An assignment is simply a link between two users in the form of a policy. Only upon policy activation, when the policy is applied, is the policy definition evaluated and calculated and the relationship confirmed. For this we need a set of steps for calculating and applying the permissions of the policy to be applied:

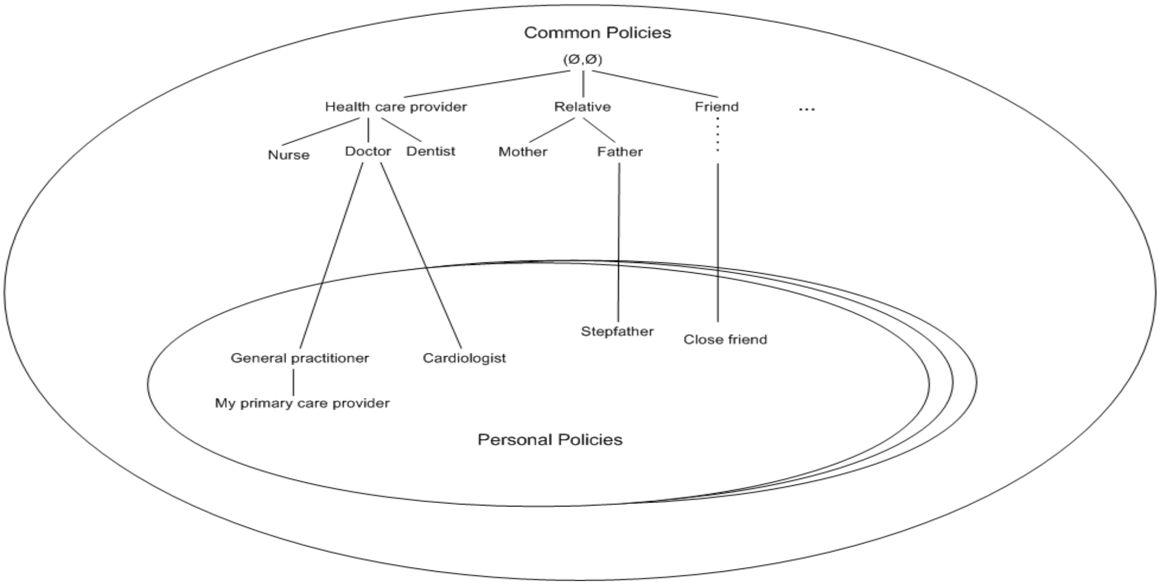

Figure 3: An example policy adaption hierarchy

1. Calculate the permissions formed by the adaption hierarchy by: first calculate adaption hierarchies formed by common policies - collapse any positive permissions together and do the same for negative permissions (two identical positive permissions results in a positive permissions, two identical negative permissions results in a negative permission etc - the presence of a negative permission at any point results in a removal of the corresponding positive permission if it is present), then take the result of this operation and do the same with any personal policies that are part of the adaption hierarchy. The result will be a set of negative permissions that is the union of all negative permissions in the adapted policies, and a set of positive permissions that is the set of all positive permissions in the adapted policies for which no corresponding negative permission exists in any of the adapted policies.
2. Calculate the intermediate policy by adding any positive permission specific to this policy for which there exists no negative permission, and by adding any negative permission that is not already part of the negative permission set.
3. Calculate the resulting policy by removing the negative permission set. The policy to be applied only consists of positive permissions. When the policy is applied, the absence of a permission is interpreted as *no access*.
4. *If* more than one policy is assigned: repeat the above steps for all assigned policies. Combine the policies by calculating the union of the permissions in all the assigned policies. Again the result is a set of only positive permissions.

This process is repeated any time a policy is applied. This may seem cumbersome and inefficient, but it affords flexibility in that any policy may be updated at any time and those changes will be reflected the next time any policy that is adapted from this one is applied. This allows great flexibility in the model.

5.5 Policy update

As stated in the requirements the personalized policies may be changed by the owner at any time. Changing or updating a policy includes adding or removing specific permissions or adding or removing policies from the policy adaption set. As stated above, the actual policy to be used is recalculated every time it is applied and as such any change will be reflected immediately. Though this approach results in a very flexible model, it also results in potential problems. If the owner changes one of her policies – is it safe to assume that she is able to grasp all the consequences of this action? Updating one policy affects all policies that are related to this one. Deleting a policy also has a cascading effect that it is difficult for the user to foresee. Potential solutions to this problem are:

- Do not allow a user to change a policy when it has been defined. This is not desirable.
- Keep a complete history of policies. If a user updates one policy that only affects policies defined after this. A copy of the old policy is kept and any pre-existing policies keeps their relationship to the old version. This is safer, but may not be what the user expects to happen.
- Only allow updates of policies that no other policy depends upon.

None of these possibilities are ideal, and work remains on how policy updates should be defined in the model.

5.6 Policy revocation

Relationships are not permanent. A patient may switch to a different doctor, visit another hospital etc. Even social relationships like family and friends may not be permanent. In this model the process of revocation is simple: it simply involves the owner removing the policy assignment. However, considering the motivating case, the consequences of revocation are not so simple. Assuming that the owner is in complete control she can deny anyone access. Depending on how the PCHR is used this may have unfortunate consequences. If the primary physician has relied on the PCHR to get information about other care the patient receives, suddenly losing access may result in lesser quality of the care he may provide. Still it is at the patient's discretion to do so. While we consider these to be valid considerations, we also consider this to be out of the scope of what can be included in a model for personalized access control but certainly an issues that needs to be resolve when the model is to be realized.

6. DISCUSSION

One of the main purposes of the PAC model is to allow the owner the power to define very specific permissions when sharing her information with someone, while at the same time preserving the flexibility provided by RBAC. Rather than having inheritance hierarchies as in standard RBAC, PAC has adaption hierarchies where it is possible to both add and subtract permissions. While this is an important property of the model, it also increases complexity by introducing the possibility for conflicting authorizations.

There are also issues that are outside the scope of this model, but nevertheless are important to mention. Many of these were summarized in an earlier paper [7] but we repeat some of them here as they are important to consider. The introduction of PCHRs is a step on the way to *patient empowerment*. Through a PCHR the patient is given complete control over who gets access to her health information. But this also implies an increased responsibility for the patient. It is important to consider how to achieve true empowerment. How do we make sure that the patient understands the consequences of every access decision she makes? Usability becomes an important feature of any implemented access mechanism based on PAC to ensure that the intentions of the model are achieved. We believe that visualization, how permissions, assignments and consequences are displayed to the user, can be used to increase the users' understanding and as such the usability of a PAC implementation. We intend to develop a set of potential visualization interfaces for PAC, in a PCHR, and perform a usability study to get feedback on the different interfaces. Such a study will yield important knowledge about the users' reactions to using PAC that will serve as valuable feedback to improving the model.

7. CONCLUSIONS

In this paper we have presented a model for personalized access control for use in a personal health record. We strongly believe that the ideas put forward here are important and bring something new to the access control field. In our future work we will focus on unresolved issues, exploring the model for various cases to make it more generic and creating a reference implementation. We will also focus on usability of PAC implementations.

8. ACKNOWLEDGMENTS

The authors would like to thank the Indivo team at the Children's Hospital Informatics Program in Boston for the opportunity to learn from, and be inspired by, their work.

9. REFERENCES

[1] Connecting for health: The personal health working group final report. Technical report, Markle Foundation, July 1 2003.

[2] American national standard for information technology - role based access control. Technical Report INCITS 359-2004, American National Standards Institute, Inc., 3 February 2004.

[3] M. A. Al-Kahtani and R. Sandhu. Rule-based rbac with negative authorization. *Computer Security Applications Conference, 2004. 20th Annual*, pages 405–415, 6-10 Dec. 2004.

[4] M. I. Kim and K. B. Johnson. Personal health records: Evaluation of functionality and utility. *J Am Med Inform Assoc*, 9(2):171–180, 2002.

[5] K. D. Mandl, P. Szolovits, and I. S. Kohane. Public standards and patients' control: how to keep electronic medical records accessible but private commentary: Open approaches to electronic patient records commentary: A patient's viewpoint. *BMJ*, 322(7281):283–287, 2001.

[6] S. Osborn, R. Sandhu, and Q. Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Trans. Inf. Syst. Secur.*, 3(2):85–106, 2000.

[7] L. Røstad. An initial model and a discussion of access control in patient controlled health records. In *The International Workshop on Privacy and Assurance (WPA-2008)*, Proceedings of the The International Conference on Availability, Reliability and Security (ARES 2008), Barcelona, Spain, 2008. IEEE Computer Society.