Purpose-based access control policies and conflicting analysis

Hua Wang, Lili Sun, Vijay Varadharajan

Purpose-based access control policies and conflicting analysis

Hua Wang[1], Lili Sun[1], and Vijay Varadharajan[2]

[1] Department of Maths and Computing, University of Southern Queensland
(wang, sun)@usq.edu.au
[2] Department of Computing, Faculty of Science
Macquarie University, Australia
vijay@ics.mq.edu.au

Abstract. This paper proposes a purpose-based framework for supporting privacy preserving access control policies and mechanisms. The mechanism enforces access policy to data containing personally identifiable information. The key component of the framework is purpose involved access control models (*PAC*) that provide full support for expressing highly complex privacy-related policies, taking into account features like purposes and conditions. A policy refers to an access right that a subject can have on an object, based on attribute predicates, obligation actions, and system conditions. Policy conflicting problems may arise when new access policies are generated. The structure of purpose involved access control policy is studied, and efficient conflict-checking algorithms are developed. Finally a discussion of our work in comparison with other access control and frameworks such as *EPAL* is presented.

Key Words: Purpose, Privacy, Access Control, Conflicts

1 Introduction

Privacy is increasing in importance since it becomes a major concern for both customers and enterprises in today's corporate marketing strategies. This raises challenging questions and problems regarding the use and protection of private messages. One principle of protecting private information is based on who is allowed to access private information and for what purpose [Agrawal et al. 2002]. For example, personal information provided by patients to hospitals may only be used with record purpose, not for advertising purpose. There must be purposes for data collection and data access. The motivations for adopting purpose based approach are 1) the fundamental policies for private information concern with which data object is used for what purposes [Byun and Li 2008] (for example, customers' age and email address are used for the purpose of marketing analysis), and 2) customers agreed data usage varies from individual to individual. Information technology provides the capability to store various types of users' information required during their business activities. Indeed, Pitofsky

[2000] showed that 97 percent of web sites were collecting at least one type of identifying information such as name, e-mail address, or postal address of consumers. The fact that the personal information is collected and can be used without any consent or awareness violates privacy for many people. This paper analyses purpose based methods to secure private information.

Data privacy is defined by policies describing to whom the data may be disclosed and what are the purposes of using the data [Abiteboul et al. 2005]. For example, a policy may specify that price of an air ticket from an agent may be disclosed, but only with "opted-in" customers, or that the price will be disclosed unless the agent has specifically "opted-out" of this default. While there is recent work on defining languages for specifying privacy policies [Schunter et al. 2003, Cranor et al. 2006], access control mechanisms for enforcing such policies have not been investigated [LeFevre et al. 2004]. Ni et al. [2007] analyzed a conditional privacy management with role based access control, which supports expressive condition languages and flexible relations among permission assignments for complex privacy policies. But many interested problems remain, for example, developing a formal method to describe and manage purposes and to automatically detect possible conflicts between access policies. As stated by Adams and Sasse (2001): "Most invasions of privacy are not intentional but due to designers' inability to anticipate how this data could be used, by whom, and how this might affect users"?

The remainder of this paper is organized as follows: Section 2 presents the motivations behind our work in this paper. Section 3 proposes a purpose based access framework which includes detailed information of purposes and access control evaluation. Section 4 provides access control policy structure and authorization models as well as illustrates the impact of generating a new access policy through examples. Section 5 describes conflict problems in access purposes and policies, and develops algorithms for detecting conflicts between purposes.. Section 6 compares the work in this paper and related previous work, the comparisons demonstrate the significance of the work in this paper. Finally, the conclusion of the paper and further work are given in Section 7.

2 Motivations

The important techniques for private information occur in distributed systems specifically tailored to support privacy policies, such as the well known P3P standard [Cranor 2006]. In particular, Agrawal et al. [2002] introduced the concept of Hippocratic databases, incorporating privacy protection in relational database systems. An important feature of their work is that it uses some privacy metadata, consisting of privacy policies and privacy authorizations stored in privacy-policies tables and privacy-authorizations tables respectively. However, they neither discussed the concepts of purpose with hierarchy structure, nor the prohibition of purpose and association of purposes and data elements. LeFevre, et al. [2004] presented an approach to enforce privacy policy in database systems. They introduced two models of cell level limited disclosure enforcement, namely table semantics and query semantics, but did not consider access control management. Ni et al. [2007] analyzed a role-based access model for purpose-based privacy protection, but their work did not consider usage access management and the conflicts between purposes in policies. The development of access technology entails addressing many challenging issues, ranging from modelling to architectures, and may lead to the next-generation of access management. This paper develops purpose based access technology for privacy violation challenges including complex policy structured models with access control.

Secure private information cannot be easily achieved by traditional access management systems because traditional access management systems focus on which user is performing what action on which data object [Wang et al. 2008b], and privacy policies are concerned with which data object is used for what purpose(s). For example, a common privacy agreement between a data collector and customers is "we use customer information for marketing purposes and to enable help us to resolve problems with services" that does not specify who can access the customer information, but only states that the information can be accessed for the purposes of marketing and customer service. Another challenge in access control policies is the conflict problem when generating new policies. For example, assume three access control policies and no conflicts between two access control policies; however it may lead to conflicts when three access policies are executed.

This paper focuses exclusively on how to specify and enforce policies for authorizing purpose-based access management using a rule-based language. We propose a comprehensive framework for purpose and data management where purposes are organized in a hierarchy. In our approach each data element is associated with a set of purposes, as opposed to a single security level in traditional secure applications. Also, the purposes form a hierarchy and can vary dynamically. These requirements are more complex than those concerning traditional secure applications. To provide sufficient functions with the framework, this paper analyses the explicit prohibition of purpose and the association of a set of purposes with access control policies. Furthermore, we discuss the conflict problems with multiple access control policies and develop algorithms for detecting and resolving conflicts. This kind of analysis for purpose-based usage control for privacy preserving has not been studied.

3 Purpose involved access control framework

This section develops a purpose based access control framework called *PACF*. *PACF* includes extended access control models and supports purpose hierarchy by introducing the intended and access purposes, and purpose associated data models. It is supposed authorization approaches in access control models to be applied for access purpose determination in database systems.

Purpose A purpose describes the reason(s) for data collection and data access [Ni et al. 2007]. A set of purposes P, is organized in a tree structure, referred to

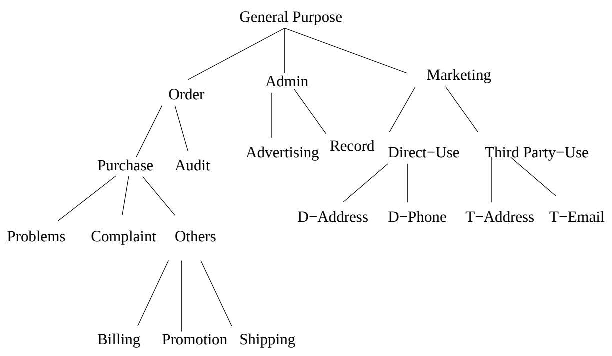

Fig. 1. Example of purpose structure

as a Purpose Tree PT, where each node represents a purpose in P and each edge represents a hierarchical relation (i.e., specialization and generalization) between two purposes. Figure 1 gives an example of a purpose tree.

Let P_i and P_j be two purposes in a purpose tree. P_i is senior to P_j (or P_j is junior to P_i) if there exists a downward path from P_i to P_j in the tree. Based on the tree structure of purposes, the partial relationships between purposes are existed. Suppose PT is a purpose tree and P is a set of purposes in PT. $Pu \in P$ is a purpose, the senior purposes of Pu, denoted by $Senior(Pu)$, is the set of all nodes that are senior to Pu. For example, *Senior(Record)* = {*Admin, General Purpose* } in Figure 1. The junior purposes of Pu, denoted by $Junior(Pu)$, is the set of all nodes that are junior to Pu. For instance, *Junior(Admin)*= {*Advertise, Record*}.

4 Access control policies

We introduce the structure of access control policy after introducing the basic concepts of purposes[Byun et al. 2005]. Let us assume a generic computer system that possesses data or resources that need to be protected from unauthorized accesses. Policies are defined to apply to this system.

Definition 4.1 An access control policy (rule) is a tuple of the form

$$(Subjects, Action, Resources, Purpose, Condition, Obligation)$$

The subjects terms identifies a user or a group who requests an action onto the resources. The action is any operation (e.g. deleting a file) to a resource in the access application. The resources term identifies a subset of objects which are normally private information that access to the objects is restricted. The purpose is selected pre-defined set of purposes that is reasons subjects intend to execute an action. The condition is a Boolean expression (i.e. a predicate) and "Obligations" are requirements that have to be followed by the subject for having access to resources. For instance, users are asked to agree to a privacy policy when installing Skype software; otherwise, the software cannot be used. We do not discuss conditions in this paper due to limited space available in this paper.

Subjects, action, and resources are the same concepts in traditional access control policies that specify who can access what with action. Purposes are applied to achieve fine-grained polices. The purpose checks for properties of the context with no intended side effects. If a side effect exists we need to consider other arguments like obligations and conditions in authorization process. We briefly discuss obligations in this paper but the detailed analysis for obligations is omitted. As we mentioned in the first section, the purpose is the reason to collect the resources and is indispensable to private access policies.

The following two examples are positive and negative authorizations, respectively. The security policy example includes two rules.

Example 2: "Hua can access purchase information for marketing purpose during working hours";

In the first rule *S = Hua, A = read, R = purchase information, P= marketing, C= 8:00am-6:00pm.* There is no obligations in the examples.

4.1 Authorization models

Definition 4.2. The *PAC* model is composed of the following components:

1. A set S of Subjects, a set D of Data, a set $Pu = \langle AIP, PIP \rangle$ of purposes (detailed AIP and PIP are in [Byun and Li, 2008]), a set A of actions, a set of O for obligations and a set of C for conditions.
2. A set of data access right $DA = \{(d, a) \mid a \in A, d \in D\}$,
3. A set of private data access right $PDR = \{(da, a, pu, c, o) \mid da \in DA, pu \in Pu, c \in C, o \in O, a \in A\}$,
4. Private data subject assignment $PDS \subseteq S \times PDR$ is a many-to-many relation that decides what subjects with which access purposes can access the private information based on authorizations.

We illustrate through an example a privacy preserving expressed with *PAC* model. Suppose that Food and Drug Administration (http://www.fda.gov/) is a web site aiming at audience that deploys its privacy policies with the purpose tree in Figure 1:

1, "Hua can read customers' PostAddress for shipping purpose".

2, "Tony can only read customers' Email address for purchase purpose if they allow to do so".

These policies are expressed as follows in *PAC* model:

P1: (Hua, (PostAdd, Read), Shipping, N/A, ϕ))

P2: (Tony, (EmailAdd, Read), Purchase, OwnerConsent='Yes', ϕ)

4.2 Policy operations

This section analyses the impact of generating new policies to an existing *PAC* model. It may have unforeseen problems while a new policy for privacy protection is raised. For example, when Tony moves to the complaint department, a new policy is defined:

5. "Tony can only read Email address of customers, for complaint purpose if they allow to do so"

The corresponding expression in *PAC* is:

P5: (Tony, (EmailAdd, Read), Complaint, OwnerConsent='Yes', ϕ).

Comparing to P2, these are two policies for Tony to access Email address for different purposes. What is the results of these two policies if combine them together? Normally, we should apply P2 for Tony to access email address for Purchase purpose and, apply P5 to access email address for Complaint purpose.

The differences in these two policies are the purposes where one is Purchase purpose while the other one is Complaint purpose. How the system will verify? Should the system verify Complaint for the access to email addresses with consent conditions? *PAC* achieves that by considering different access policies as linked by a conjunction.

That is, if a user U wants to access right a on data d for purpose Pu, all access polices of U related to $((d, a), Pu)$ must be checked. U can read the d if there exists at least one policy and U can satisfy all purposes in all policies. If a new access policy is related to the same user, same data, same right and same conditions of some existed private policies, it is not used to relax the access situations but to make the access stricter. If privacy officers want to relax the access environments, they can do so by revising the existed access policies instead of creating a new one.

Suppose two private access policies in *PAC*: $(u_1, (d_1, r_1), pu_1, c_1, \phi)$ and $(u_1, (d_1, r_1), pu_2, c_1, \phi)$, can we simply replace them with a new one as $(u_1, (r_1, d_1), pu_1 \wedge pu_2, c_1, \phi)$? Consider $P2$ and $P5$, we have the following policy:

P6: (Tony, (EmailAdd, Read), Complaint $\wedge$ Purchase, OwnerConsent = 'Yes', ϕ).

From the purpose hierarchy structure in Figure 1, $Complaint \wedge Purchase = Complaint$ since *Complaint* is junior to purpose *Purchase*. Translating $P6$ into plain English, we obtain "Tony can read customers' Email address for Complaint purpose if the customers agree to do so". The translating is not correct since something is lost. Tony cannot access email addresses, for purposes of Problem solving and Other purchase purposes which are not included. The reason for this is the context variable purchase purpose in P5. The variable purchase purpose separates the values of order into three disjoint sets: Complaint, Problem solving and Others not included in the first two purposes. P2 thus applies to all three kinds of customers, while P5 only applies to email addresses for Complaint purpose. Simply combining purposes in P2 with purposes in P5 actually removes all purposes except Complaint purpose for access email addresses.

The notion of splitting context variables is required to analyse this problem [Ni, et al. 2007].

Definition 4.2. A splitting context variable (*SCV*) is a context variable that satisfies the following conditions.

1. A *SCV* is related to purpose information.
2. The values of an *SCV* partition purposes into disjoint sets.

3. A *SCV* is not used to represent information about consent.

Based on the *SCV* definition, *Order* is *SCV*, whereas *Admin* and *Direct-Use* are not since the joint sets of *Advertising* and *Record*, *D-Address* and *D-Phone* are not empty. The notion of *SCV* is important and is used in the analysis of the paper. We are now able to give an answer to the aforementioned question: only if both pu_1 and pu_2 do not involve *SCV*, or the *SCV* that they involve have the same values, they could be safely rewritten into $pu_1 \wedge pu_2$.

Consider the following two access policies:

P7: (Tony, (EmailAdd, Read), Complaint, OwnerConsent='Yes', ϕ)

P8: (Tony, (EmailAdd, Read), N/A, OwnerConsent='Yes', ϕ).

$P7$ and $P8$ can be revised as:

P9: (Tony, (EmailAdd, Read), Complaint, OwnerConsent='Yes', ϕ).

Similarly, the following two access policies:

P10: (Tony, (EmailAdd, Read), Shipping, OwnerAge $\leq$ 13, ϕ)

P11: (Tony, (EmailAdd, Read), Record, OwnerAge $\leq$ 13 ϕ)

P12: (Tony, (EmailAdd, Read), Shipping $\wedge$ Record , OwnerAge $\leq$ 13, ϕ)

P12 is equivalent to $P10$ and $P11$. We now rewrite $P2$ and $P5$ as following policies:

P13: (Tony, (EmailAdd, Read), Shipping $\cup$ Billing $\cup$ Problemsolving $\cup$ Promotion, OwnerConsent='Yes', ϕ)

P14: (Tony, (EmailAdd, Read), Complaint, OwnerConsent='Yes', ϕ)

It is easy to understand P13 and P14 rather than P2 and P5. $\cup$ means "or" in the example. We do not have obligations in the discussion above. What may happen if there are obligations? Consider the following example:

P15: (Tony, (EmailAdd, Read), Complaint, OwnerConsent='Yes', NotifybyPhone)

P16: (Tony, (EmailAdd, Read), Purchase, OwnerConsent='Yes', NotifybyEmail)

Intuitively, P15 is fine for Tony reading customers' email address for Complaint purpose. This means that the phone activity should be invoked for Complaint purpose when accessing customers' data for Purchase purpose by notified by Email. Therefore, their equivalent forms are:

P17: (Tony, (EmailAdd, Read), Complaint, OwnerConsent='Yes', NotifybyPhone and NotifybyEmail)

P18: (Tony, (EmailAdd, Read), Shipping $\cup$ Billing $\cup$ Problemsolving $\cup$ Promotion, OwnerConsent='Yes', NotifybyEmail)

In summary, a private data access request related to user u, data d, access right a, purpose Pu is authorized only if all access policies related to $(u, (r, d), Pu)$ are satisfied. If so, obligations in all applicable policies are invoked after the access request.

5 Conflicting algorithms

In the section, we discuss the various cases of conflicting policies in *PAC* model. It is not easy to comply with complex security and privacy policies, especially in large enterprises. The more complex a security policy is, the larger is the probability that such policy contains inconsistent and conflicting parts.

Consider the following policies:

P19: (Christine, (Read, OrderInfor), Shipping, Time=5PM-11PM, ϕ)

P20: (Christine, (Read, OrderInfor), Problem solving, Time=5PM-11PM, ϕ)

These two policies do not conflict with each other because P19 and P20 actually work on different purposes. The *SCV Order* used in these two policies as purposes with different values. It is called incomparable policies because they have incomparable purposes, that is, a *SCV* exists which has two disjoint value sets in the two purposes.

Definition 5.1. Let pu_i and pu_j be two purposes in two access control policies. We say that pu_i and pu_j are incomparable purposes if there exists a common *SCV* that has disjoint value sets in purposes pu_i and pu_i. Otherwise, we say that pu_i and pu_j are comparable purposes, written as $pu_i \approx pu_j$.

Consider the following two permission assignments which include comparable purposes:

P21: (Christine, (Read, OrderInfor), Purchase, Time = 9AM-5PM, ϕ)

P22: (Christine, (Read, OrderInfor), Billing, Time = 9AM-5PM, ϕ)

Because P21 allows data access during 9AM - 5PM with Purchase purpose and P22 allows data access during in the same time with Billing purpose, a data request occurs during 9AM - 5 PM with Billing purpose could be authorized. These two policies are compatible because they have compatible purposes: the intersection of value sets of context variable *Order* in different access policies is not empty.

Besides compatible purposes, we may have conflicting purposes.

P23: (Christine, (Read, OrderInfor), purchase, Time = 5PM-11PM,ϕ)

P24: (Christine, (Read, OrderInfor), audit, Time = 5PM-11PM, ϕ).

P23 specifies that Christine is authorized to access order information for Purchase during 5PM-11PM, whereas P24 allows partners' access with *Audit* during 5PM-11PM. Hence, when data access request is issued, the access purpose could not be both purchase and audit. Therefore, any data request will be denied according to these two access policies. These two permission assignments conflict with each other because they have conflicting purposes, that is, no value of the context variable *Order* could satisfy both purposes.

Definition 5.2. Let pu_i and pu_j be two comparable purposes in two access policies. We say that pu_i and pu_j are conflicting purposes if there exists at least one common context variable in pu_i and pu_j that has disjoint value sets, written as $pu_i \asymp pu_j$. Otherwise, we say that pu_i and pu_j are compatible purposes.

Consider the following access policies which include conflicting obligations:

P25: (Christine, (Read, OrderInfor), purchase, N/A, Notify())

P26: (Christine, (Read, OrderInfor), purchase, N/A, Notify(Opt-out))

Once a data request is authorized, the system does not know which obligation should be executed (either Notify or Notify with Opt-out); therefore P25 conflicts with P26.

We denote the fact that two obligations o_i and o_j conflict as $o_i \asymp o_j$.

Based on aforementioned definitions and examples, we give the definition of conflicting access policies.

Definition 5.3. Let $Pi = (ui, (ri, di), pui, ci, oi)$ and $Pj = (uj, (rj, dj), puj, cj, oj)$ be two privacy-sensitive data access policies. We say that Pi and Pj are conflicting if one of the following two conditions holds:
$(ui = uj) \wedge (ri = rj) \wedge (di = dj) \wedge (ci = cj) \wedge (pui \asymp puj)$
$(ui = uj) \wedge (ri = rj) \wedge (di = dj) \wedge (ci = cj) \wedge (pui \approx puj) \wedge (oi \asymp oj)$

In *PAC*, conflicting access policies should be detected and one of them should be removed to prevent ambiguities when enforcing access policies.

5.1 Detecting algorithms

Conflicting policies detection is important in order to guarantee the consistency of access control policy. In this section, we present algorithms to detect conflicts between purposes and to check conflicts in access control policies. The key point of the algorithm is that we first sort context variables used in conditions according to their name, then make a disjoint test for the value sets for a variable in the various conditions.

Algorithm 1 Purpose-Conflict(pu1, pu2)
Require: *pu1* and *pu2* are two purposes applied in two access control policies
Outcomes: True //Purposes have conflicts
False //Otherwise

```
1: $pul_1$: Sort context variables used in $pu1$ according to their name
2: $pul_2$: Sort context variables used in $pu2$ according to their name
3: for(integer $i$ = 1 to $|pul_1|$)
4:    { for(integer $j$ = 1 to $|pul_2|$)
5:      { if $pul_1[i].name = pul_2[j].name$ // Common context variable
6:      then
7:        { if $pul_1[i].SCV = True$ // $pul_1[i]$ is a SCV
8:          {if disjointTest($pul_1[i].value, pul_2[j].value$) = 'False' // $pul_1[i].value$ and
9:          //$pul_2[j].value$ have joint value sets, no conflicts between $pul_1[i]$ and $pul_2[j]$
10:           then j++ //check the next purpose in $pu2$
11:           else
12:           Return True //Conflict purposes }
13:         else j++ //check the next purpose in $pu2$ }
14:       else j++ }
15:   i++ //check the next purpose in $pu1$
16: Return result
```

Based on the Purpose-Conflict algorithm, the access control policy detection algorithm is given below.

Algorithm 2 Policy-Conflict(po1, po2)
Require: po_1 and po_2 are two access control policies
Outcomes: True //Policies have conflicts

```
            False //Otherwise
1: if $po_1.s \neq po_2.s$ or $po_1.d \neq po_2.d$ or $po_1.r \neq po_2.r$ or $po_1.c \neq po_2.c$ or $po_1.o \neq po_2.o$, then
2: return False
3:    end if
4:      { if Purpose-Conflict($po_1.pu, po_2.pu$) = True
5:      //Checking conflicts between two purposes in two policies
7:      return True // policies conflict
8:      else // $po_1.pu \approx po_2.pu$
9:        {if { SCV-Disjoint($po_1.o, po_2.o$) = True //obligations are comparable
10:         then
11:           {if Obligation-Conflict($po_1.o, po_2.o$)= True
12:           return True // Obligations conflicts
13:           else return False //no conficts in policies }
14:         else // SCV-Disjoint($po_1.o, po_2.o$) = False, Obligation incomparable
15:         return False // No conflicts in policies}
16:       }
```

Based on Algorithm 1, 2 and the structure of access purpose and policy, we can further develop algorithms with *SQL* to support the purpose and policy management approach presented in this paper. The detailed methods with *SQL* are omitted due to the length of the paper.

6 Comparisons

We present a brief comparison of the purpose involved access model *PAC*against other related work. The closely related works to this paper are privacy-aware role-based access control [Ni et al. 2007] and the enterprise privacy authorization language (*EPAL*)[Schunter, et al. 2003].

Ni et al [2007] introduced a family of models that extend the well known *RBAC* model in order to provide full support for expressing highly complex privacy-related policies, taking into account features like purposes and obligations. The models include the *Core P-RBAC* model, *Hierarchical P-RBAC* model, *Conditional P-RBAC* and *Universal P-RBAC*. Their work is different from ours in three aspects. First, their paper is focused on the conditions and their relationships in role-based access control. By contrast, our work has analyzed the purpose hierarchy structure in access control policies in usage access control model. Second, the conflicts between two P-RBAC permission assignments discussed in their paper are based on conditions. They neither analyze the access purpose structure nor the impact of adding a new access policy with different purposes. By contrast, our work has analyzed purpose hierarchical structure and the impact of adding new access control policies, specially the conflicting problem between three purposes.

EPAL [Schunter, et al. 2003] is a formal language for writing enterprise privacy policies to govern data handling practices in IT systems according to fine-grained positive and negative authorization rights. It concentrates on the core privacy authorization while abstracting data models and user-authentication from all deployment details such as data model or user-authentication. An *EPAL* policy defines lists of hierarchies of data-categories, user-categories, and purposes, and sets of (privacy) actions, obligations, and conditions. Purposes model the intended service for which data is used (e.g., processing a travel expense reimbursement or auditing purposes). Compared to *EPAL*, *PAC* has the following major differences. First, one of the important design criteria of *PAC* is to unify privacy policy enforcement and access control policy enforcement into one access control model. By contrast, EPAL is designed independently from any access control model. Second, the conflicting policies problem was not introduced and analyzed in *EPAL*; hence shortcoming exists during answering data access request [Barth et al. 2004], but *PAC* supports conflict detection to guarantee that no conflicts arise in the procedures of generating new policies, thus preventing the disclosure of private information. Third, the basic ideas of purpose in *PAC* are borrowed from *EPAL*, the purposes in EPAL represent reasons of data collection without further discussion such as conflicts from a privacy perspective; by contrast purposes in *PAC* have rich analysis and conflict algorithms.

7 Conclusions and future work

This paper has discussed purpose-based access control policies with conditions and obligations. We have studied the access control framework but also the structure of access policies including subjects, access actions, resources, purposes and obligations. We have also analyzed the impact of adding new policies and the conflicts that they can lead to. Algorithms have been developed to help a system to detect and solve the problems. The work in this paper has extended previous work significantly in several aspects, for example, purpose involved access control, access control policies and generating a new access policy without conflicts.

The research for purpose involved access control policies is still in its infancy and much further work remains to be done. There could exist redundant access policies in *PAC*. For instance, P7 is redundant with respect to P8. Formal definitions of the redundancy need to be developed and solutions for addressing them are possible avenues for our future work.

References

1. Abiteboul, S. and Agrawal, R. 2005. The *Lowell* database research self-assessment. *Communications of the ACM.* 48 (5),111–118.
2. Agrawal, R., Kiernan, J., Srikant, R. and Xu, Y. 2002. Hippocratic databases. *Proc. 28th Int'l Conf. on Very Large Data Bases.* Hong Kong, China, 143–154.
3. Adams, A., Sasse, A., 2001. Privacy in Multimedia Communications: protecting users, not just data. *People and Computers XV - Interaction Without Frontiers.* Joint Proceedings of HCI2001 and ICM2001, pp. 49-64.

4. Barth A., Mitchell, J.C. and Rosenstein J. 2004. Conflict and combination in privacy policy languages. *Proceedings of the ACM workshop on Privacy in the electronic society*, pages 45-46.
5. Bertino, E., Samarati, P., and Jajodia, S. 1997. An Extended Authorization Model for Relational Databases. *TKDE.* 9(1), 85-101.
6. Bertino, E., Byun, J.-W. and Li, N. 2005. Privacy-preserving database systems. *Lecture Notes in Computer Science.* Springer Berlin, Heidelberg, 178–206.
7. Bonatti, P., Damiani, E., de Capitani, S., and Samarati, P. 2001. A Component-Based Architecture for Secure Data Publication. *Proceedings of the 17th Annual Computer Security Applications Conference.* IEEE Computer Society, pp, 309.
8. Bonatti, P., Damiani, E., De Capitani di Vimercati, S., and Samarati, P. 2001. An access control model for data archives. *Proceedings of the 16th international Conference on information Security: Trusted information: the New Decade Challenge.* Kluwer Academic Publishers, Norwell, MA, 261-276
9. Byun, J.-W., Bertino, E. and Li, N. 2005. Purpose based access control of complex data for privacy protection. *Proceedings of the 10th ACM symposium on Access control models and technologies.* NY, USA, 102–110.
10. Byun, J. and Li, N. 2008. Purpose based access control for privacy protection in relational database systems. *The VLDB Journal.* 17 (4), 603-619.
11. Clifton, C. 2000. Using sample size to limit exposure to data mining. *Journal of Computer Security.* 8(4), 281–307.
12. Cranor L. et al. 2006. The platform for privacy preferences 1.1 (P3P) specification. *W3C Working Group.*
13. LeFevre, K., Agrawal, R., Ercegovac, V., Ramakrishnan, R., Xu, Y. and DeWitt, D. 2004. Limiting disclosure in hippocratic databases. *Proceedings of the 13th VLDB conferenc.* 108–119.
14. Ni, Q., Lin, D., Bertino, E. and Lobo, J. 2007. Conditional privacy-aware role based access control. *ESORICS*, 72–89.
15. Ni, Q., Trombetta, A., Bertino, E., and Lobo, J. 2007. Privacy-aware role based access control. *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies.* France, 41-50.
16. Li M., Sun X., Wang H. and Zhang Y., Optimal Privacy-aware Path in Hippocratic Databases, *The 14th International Conference on Database Systems for Advanced Applications(DASFAA2009)*, April 21-23, 2009, Brisbane, Australia.
17. Li, N., Yu T. and Anton, A. 2003. A semantics-based approach to privacy languages. *Technical Report, Nov. 2003.* TR 2003-28.
18. Pitofsky, R. et al. 2000. Privacy online: Fair information practices in the electronic marketplace, a report to congress, *Federal Trade Commission.*
19. Schunter M. et al. 2003. The enterprise privacy authorization language (epal 1.1). *W3C Working Group.*
20. Sweeney, L. 2002b. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst..* 10(5), 571–588.
21. Wang, H., Cao, J. and Zhang, Y. 2008b. Access control management for ubiquitous computing. *Future Generation Computer Systems journal.* 24, 870-878.
22. Zhu H. and Lu K. 2007. Fine-Grained Access Control for Database Management Systems. *Data Management. Data, Data Everywhere.* 215-223.