4-2014

Building access control policy model for Privacy Preserving and Testing Policy Conflicting Problems

Elisa Bertino
Purdue University, bertino@cs.purdue.edu

Hua Wang
Victoria University, Victoria, Austrailia, hua.wang@vu.edu.au

Lili Sun
Victoria University, Victoria, Austrailia, lili.sun@vu.edue.au

Journal of Computer and System Sciences 80 (2014) 1493–1503

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Building access control policy model for privacy preserving and testing policy conflicting problems

Hua Wang [a], Lili Sun [a], Elisa Bertino [b]

[a] Centre for Applied Informatics, Victoria University, Victoria, Australia
[b] Purdue University, West Lafayette, United States

ARTICLE INFO

Article history:
Received 8 January 2013
Received in revised form 20 August 2013
Accepted 10 April 2014
Available online 16 April 2014

Keywords:
Purpose
Privacy
Access Control
Conflicts

ABSTRACT

This paper proposes a purpose-based access control model in distributed computing environment for privacy preserving policies and mechanisms, and describes algorithms for policy conflicting problems. The mechanism enforces access policy to data containing personally identifiable information. The key component is purpose involved access control models for expressing highly complex privacy-related policies with various features. 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 that are possible to be conflicted to existing policies. As a result of the policy conflicts, private information cannot be well protected. The structure of purpose involved access control policy is studied, and efficient conflict-checking algorithms are developed and implemented. Finally a discussion of our work in comparison with other related work such as EPAL is presented.

1. Introduction

Privacy preserving is increasing in its importance since privacy 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, especially for context-aware web service [6]. One principle of protecting private information is based on who is allowed to access private information and for what purpose [2]. 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 [20] (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 [21] showed that 97 percent of web sites were collecting at least one type of identifying information such as name, home address, 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 analyzes 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 [1]. For example, a policy may specify that price of an air ticket from an agent may be disclosed, but only with

E-mail addresses: wang@usq.edu.au (H. Wang), sun@usq.edu.au (L. Sun), bertino@cs.purdue.edu (E. Bertino).

http://dx.doi.org/10.1016/j.jcss.2014.04.017
0022-0000/

"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 [22,11], access control mechanisms for enforcing such policies have not been investigated [16]. Ni et al. [19] 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 Al-Harbi and Osborn [4] and Adams and Sasse [3]: "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"?

Access control is significant when disclosing private information in web service [14]. The importance of privacy has been recognized for a long time, but the concept has not been supported in traditional access models, especially purpose based access control systems. A security officer has to check privacy policies if an access is required. Furthermore, administrators are prone to making mistakes when they generate new access policies to access sensitive data [7]. Such an approach significantly increases the management efforts in distributed environments because of the various privacy requirements and the continuous involvement from security officers. This paper bridges the gap between private information protecting technology and access control models. We start from building a purpose-based access framework and analyzing the conflicts between purposes in access control policies.

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. The implementation of the conflicting algorithms is described in Section 6. Section 7 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 8.

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 [27,11,13]. In particular, Agrawal et al. [2] 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. [15] 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. Li et al. [16] devised generalization boundary techniques to maximize data usability while, minimizing disclosure of privacy. Inspired by the fact that the permissible generalization level results in a much finer level access control, the authors proposed a privacy-aware access control model in web service environments and also analyzed an access process management through a trust-based decision and ongoing access control policies. However, the concept of purpose was missed. Ni et al. [19] 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 modeling 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.

Privacy violations may happen when data are released to third parties [2]. Data once released are not any longer under the control of the organizations owning them, and the data owners are not able to control the way data are used. The most common approach to address the privacy of released data is to modify the data by removing all information that can directly link data items with individuals [24]. It is important to note that simply removing identity information, like names or social-security numbers, from the released data may not be enough to anonymize the data. Many examples show that even when such information is removed from the released data, the remaining data combined with other information sources may still link the information to the individuals it refers to [23]. Sweeney [25] proposed approaches based on the notion of k-anonymity as solutions of the problem. Another secure private information techniques such as density-based clustering algorithms happens in the context of data mining [18].

Data mining techniques are today very effective. Thus even though a database is sanitized by removing private information, the use of data mining techniques may allow one to recover the removed information. These techniques are based on modifying or perturbing the data in some way; for example, techniques specialized for privacy preserving mining of association rules modify the data so to reduce the confidence of sensitive association rules [12]. A problem common to those techniques is represented by the quality of the resulting data; if data undergo too many modifications, they may not be useful any longer [10].

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 [28], 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

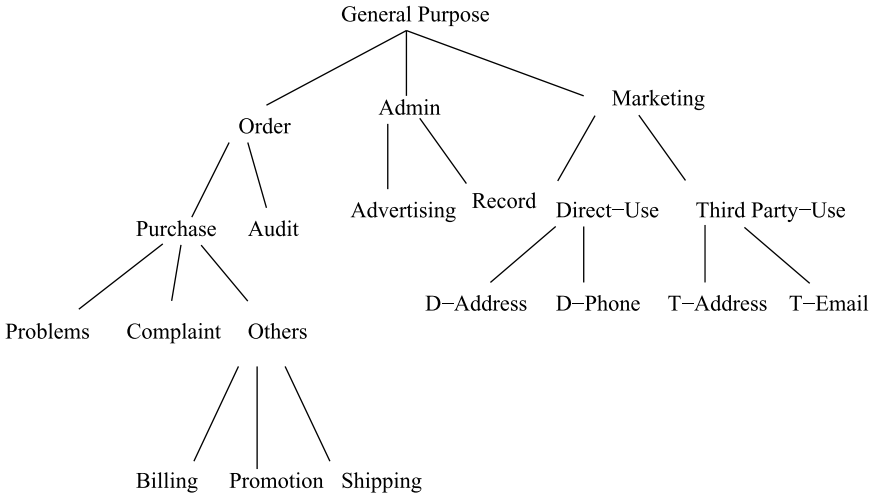

Fig. 1. Example of purpose structure.

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 [17]. To provide sufficient functions with the framework, this paper analyzes 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 [19]. A set of purposes P, is organized in a tree structure, referred to 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. Fig. 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 Fig. 1. The junior purposes of Pu, denoted by $Junior(Pu)$, is the set of all nodes that are junior to Pu. For instance, $Junior(Admin) = \{Advertise, Record\}$.

We design an access control model by adding purposes and policy language, and discuss the details of the access purpose authorization and verification based on the model. Intuitively, an access to a specific data element should be allowed if the allowed purposes for the data, stated by the privacy policies, include or imply the purpose of the data access. Access purpose authorizations are granted to users based on the access purpose on the data, obligations and conditions. Authorizations approaches in access control such as *pre-Authorizations model* and *ongoing-Authorizations model* have already been introduced [26], and access purpose authorizations in access control policies are analyzed in this paper.

4. Access control policies

We introduce the structure of access control policy after introducing the basic concepts of purposes [8]. Policies are defined to apply to this system. Let us assume a generic computer system that possesses data or resources that need to be

protected from unauthorized accesses. Privacy preserving is achieved by through authorization models and policy operations in the designed access control policies.

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 accept the agreement of 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 1: "Hua can access purchase information for marketing purpose during working hours";
Example 2: "Christine cannot update phone numbers for record purpose anytime".

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. The second example with negative authorization, $S = Christine$, $A = update$, $R = phone\ number$, $P = record$, $C = anytime$.

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 [9]), 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.

In what follows we provide additional details on the purpose involved language of *PAC* model and elaborate on conflicts among purposes and obligations. To simplify the purpose involved authorization models, we assume that $PIP = \phi$, and then $Pu = AIP$.

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 Fig. 1:

1) Subjects = {Hua, Tony, Christine, Den},
2) Action = {Read, Update, Delete},
3) Data = {OrderInfo, HomePhone, PostAdd, EmailAdd},
4) Purpose = {Order, Complaint, Billing, Shipping, ProblemSolving, Others}.

The following privacy policies:

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".
3. "Christine may read customers' order information for Billing purpose; and customers will be informed by Email".
4. "Den can read customers' Home Phone for Problem solving if it is approved by Hua".

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)

P3: (Christine, (OrderInfor, Read), Billing, N/A, Notify(ByEmail))
P4: (Den, (HomePhone, Read), Problemsolving, 'Approved by Hua', N/A)

4.2. Policy operations

This section analyzes 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 Fig. 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 analyze this problem [19].

Definition 4.3. A splitting context variable (*SCV*) is a context variable that satisfies the following conditions.

1. An *SCV* is related to purpose information.
2. The values of an *SCV* partition purposes into disjoint sets.
3. An *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 ≤ 13, ϕ)
P11: (Tony, (EmailAdd, Read), Record, OwnerAge ≤ 13 ϕ)
P12: (Tony, (EmailAdd, Read), Shipping $\wedge$ Record, OwnerAge ≤ 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, an *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–5PM 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.

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 an SCV
8: {if disjointTest($pul_1[i].value, pul_2[j].value$) = 'False' $pul_1[i].value$ and $pul_2[j].value$ have joint value sets,
no conflicts between $pul_1[i]$ and $pul_2[j]$
9: then j++ //check the next purpose in *pu*2
10: else
11: Return True //Conflict purposes }
12: else j++ //check the next purpose in *pu*2 }
13: else j++ }
14: i++ //check the next purpose in *pu*1
15: Return result

Based on the Purpose-Conflict algorithm, the access control policy detection algorithm is given below. The idea of the algorithm is to test the purpose conflicts first, if so the policies are conflict. Otherwise, check the obligations to determine if or not the policies are conflict.

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 $po_1.c \neq po_2.c$, 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
6: return True //purposes conflict
7: //policies conflict
8: else //$po_1.pu \approx po_2.pu$
9: {if $(po_1.o \cap po_2.o) = \phi$ //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 conflicts 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 Algorithms 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.

6. Experimental results

This section presents the implementation of the access control policy algorithms with Microsoft Visual Studio technology. The reason of using Microsoft Visual Studio technology is that we do not need to worry about the data structure including attributes in each access control policy. It is easy to add attributes to the existing *policy* table when another access policy is required to join or create in future. This is a web-based project implemented in *XAMPP* (http://www.apachefriends.org/en/xampp.html) environment in windows platform using *MySql* database and *Apache* web server. Due to the open source *MySQL* database with high performance, high reliability and ease of use, it is used to store the information of subjects, actions, resources, purposes and obligations in policy in the implementation. We store the structure of policy, purpose hierarchy, resources and so on. *Apache* was the first viable alternative to the Netscape Communications Corporation web server, and since has evolved to dominate other web servers in terms of functionality and performance, making applications easily portable to all of the operating systems on which Google Chrome runs. With Microsoft Visual Studio, it is able to go to the next level with *html5*.

The implementation of the access control policy algorithms includes many components, for example:

1) The structure of database including access control policy, resource and purpose
2) Conflicts of purposes, obligations and policies
3) Conditions and obligations in policies.

Clients are requiring using a modern web browser such as Google Chrome, Mozilla Firebox, Internet Explorer 6 or over and enable cookies. The computer must have an Internet connection in order to be able to access the system.

Database Design

The database to implement the *PAC* model consists of many tables such as *Policy*, *Resource* and *Purpose*. For example, the policy table named *policy.mdf* is defined in the *PAC* model which is a tuple of form (*Subject, Action, Resources, Purpose, Condition, Obligation*). Fig. 2 below shows the definition of the *Policy* Table.

User Interface

In order to make the implementation more convenient we developed a graphical user interface which interacts with the procedures of creating and removing policies. The graphical user interface is illustrated in Fig. 3. This interface was developed using Microsoft Visual Studio 2010 Ultimate and *MySql* database is used to initiate the generating access control policy instead of typing the above procedure call. This implementation is convenient for administrators since they only need to define the purpose hierarchy and obligation structure.

Fig. 3 shows the page of all policies defined in database in a data grid. They can be modified by *edit* and *delete*. To access the resource we have to enter policy *Id* in the text box given upper side of data grid and click on *Submit* button. To add new policy we need to click on *Add Policy* button as below.

Column Name	Data Type	Allow Nulls
PolicyId	bigint	☐
Subject	varchar(50)	☐
Action	varchar(50)	☐
Resourse	varchar(50)	☐
Purpose	varchar(50)	☐
Obligation	varchar(50)	☐

Fig. 2. Policy structure.

Enter policy Id:

Submit

	PolicyId	Subject	Action	Resourse	Purpose	Obligation
Edit Delete	16	Christine	r	OrderInformation	Billing	NA
Edit Delete	17	Christine	r	OrderInformation	Billing	NAT
Edit Delete	19	Hua01	r	Customers	Audit	yes
Edit Delete	20	Hua01	r	Customers	Research	yes
Edit Delete	22	Hua01	r	Customers	Post	no
Edit Delete	23	Hua01	r	Customers	Purchase	no
Edit Delete	24	Hua01	r	OrderInformation	Purchase	no
Edit Delete	25	Hua	r	Customers	Purchase	yes
Edit Delete	26	Hua	w	Customers	Research	NA

AddPolicy

Fig. 3. Creating policy.

Comparable Policy

Subject: Christire

Action: r

Resource: OrderInformation

Purpose: Billing

Obligation: NA

Confirm

Fig. 4. Comparable policy.

Comparable policy

If a new policy is created which consists of the data given below, the new policy is comparable to the existing *policy* 16 which has same Subject, Action, Resource and Obligation except purpose with *Purchase* and here purpose is *Billing* in Fig. 4. Due to the comparable property it will show a message of comparable policy and add the new policy to the database.

Conflict of purposes

When a new policy is planed to add which has data like given below, the new policy is now compared to the *policy* 16 which has same Subject, Action, Resource and Obligation and purpose is *Purchase* and here purpose is *Audit*, due to they are incompatible it will give us message of conflict of purpose and the new policy is not able to add to the database as shown in Fig. 5.

Conflict of obligations

It is quite similar to the conflict of purposes when the implementation deals with the obligation conflicts. When a new policy which has data given below, it is compared to the existing *policy* 16 which has same Subject, Action, Resource and Purpose but Obligation is *NA* and here Obligation is *NAT*. Due to they are conflicted it will give us message of conflict of obligation and it is not able to add to the database as shown in Fig. 6.

The advantages of the implementation are 1) providing a user interface for administrators to manage access control policies with various purposes and obligations since users can create and delete policy from the database without technical

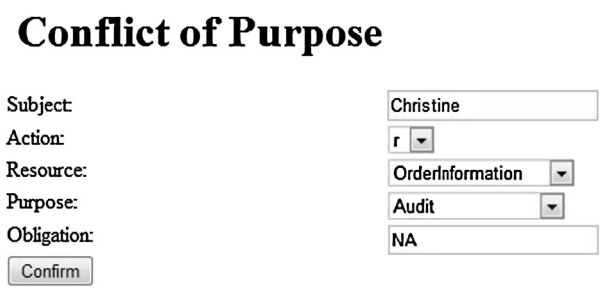

Fig. 5. conflict of purposes.

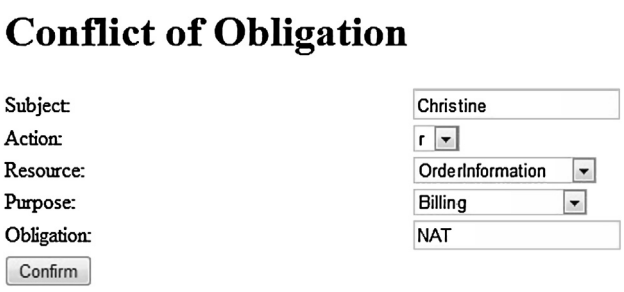

Fig. 6. Creating policy with conflicting obligation.

support which is efficient for system organizers; 2) The system provides a solution for conflicting policy problem, it supports not only the conflicts of purpose, but also the conflicts of obligations. 3) Implementation with Microsoft Visual Studio technology which is easy to manage life targeting an increasing number of platforms and technologies in future. However, we do not analyze generalized temporal constraints and the workflow of the policy creating processes, we suppose to complete them in our future work.

7. 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 [19], the enterprise privacy authorization language (*EPAL*) [22] and a conditional role-involved purpose-based access control model [14].

Ni et al. [19] 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 [22] 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 [5], 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.

The paper [14] proposed a privacy preserving access control which is based on variety of purposes. Conditional purpose is applied along with allowed purpose and prohibited purpose in the model. The structure of conditional purpose-based access control model is defined and investigated through dynamic roles. An algorithm is developed to achieve the compliance

computation between access purposes and intended purposes and is illustrated with Role-based access control (RBAC) in a dynamic manner to support conditional purpose-based access control. However, the paper did not analyze the structure of access control policy, nor the associated access control models, access purposes, obligations and conflicts between access purposes and between access control policies, but instead discussed how to extend traditional access control models to a further coverage of privacy preserving in data mining atmosphere.

8. Conclusions and future work

This paper has discussed purpose-based access control policies with conditions and obligations in distributed computing environments. 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 and to help a system to detect and solve the problems. Furthermore, the experimental results demonstrate the practicality and performance of the algorithms. 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] S. Abiteboul, R. Agrawal, The Lowell database research self-assessment, Commun. ACM 48 (5) (2005) 111–118.
[2] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu, Hippocratic databases, in: Proc. 28th Int'l Conf. on Very Large Data Bases, Hong Kong, China, 2002, pp. 143–154.
[3] A. Adams, A. Sasse, Privacy in multimedia communications: protecting users, not just data, in: People and Computers XV – Interaction Without Frontiers. Joint Proceedings of HCI2001 and ICM2001, 2001, pp. 49–64.
[4] A. Al-Harbi, S. Osborn, Mixing privacy with role-based access control, in: Proceedings of The Fourth International Conference on Computer Science and Software Engineering, Montreal, Quebec, Canada, May 16–18, 2011, pp. 1–7.
[5] A. Barth, J.C. Mitchell, J. Rosenstein, Conflict and combination in privacy policy languages, in: Proceedings of the ACM Workshop on Privacy in the Electronic Society, 2004, pp. 45–46.
[6] E. Bertino, P. Samarati, S. Jajodia, An extended authorization model for relational databases, IEEE Trans. Knowl. Data Eng. 9 (1) (1997) 85–101.
[7] E. Bertino, J.-W. Byun, N. Li, Privacy-Preserving Database Systems, Lect. Notes Comput. Sci., Springer, Berlin, Heidelberg, 2005, pp. 178–206.
[8] J.-W. Byun, E. Bertino, N. Li, Purpose based access control of complex data for privacy protection, in: Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, NY, USA, 2005, pp. 102–110.
[9] J. Byun, N. Li, Purpose based access control for privacy protection in relational database systems, VLDB J. 17 (4) (2008) 603–619.
[10] C. Clifton, Using sample size to limit exposure to data mining, J. Comput. Secur. 8 (4) (2000) 281–307.
[11] L. Cranor, et al., The Platform for Privacy Preferences 1.1 (P3P) Specification, W3C Working Group, 2006.
[12] F. Folino, C. Pizzuti, Combining Markov models and association analysis for disease prediction, in: Proceedings of the Second International Conference on Information Technology in Bio- and Medical Informatics, France, 2011.
[13] X. Huang, Y. Xiang, A. Chonka, J. Zhou, R. Deng, A generic framework for three-factor authentication: preserving security and privacy in distributed systems, IEEE Trans. Parallel Distrib. Syst. 22 (8) (August 2011) 1390–1397.
[14] M. Kabir, H. Wang, E. Bertino, A conditional role-involved purpose-based access control model, J. Organ. Comput. Electron. Commer. 21 (1) (2011) 71–91.
[15] K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu, D. DeWitt, Limiting disclosure in hippocratic databases, in: Proceedings of the 13th VLDB Conference, 2004, pp. 108–119.
[16] M. Li, X. Sun, H. Wang, Y. Zhang, J. Zhang, Privacy-aware access control with trust management in web service, World Wide Web 14 (4) (July 2011) 407–430.
[17] N. Li, T. Yu, A. Anton, A semantics-based approach to privacy languages, Technical Report, Nov. 2003. TR 2003-28, 2003.
[18] J. Liu, J. Huang, J. Luo, L. Xiong, Privacy preserving distributed DBSCAN clustering, in: Divesh Srivastava, Ari Ismail (Eds.), Proceedings of the Joint EDBT/ICDT Workshops, EDBT-ICDT '12, ACM, New York, NY, USA, 2012, pp. 177–185.
[19] Q. Ni, A. Trombetta, E. Bertino, J. Lobo, Privacy-aware role based access control, in: Proceedings of the 12th ACM Symposium on Access Control Models and Technologies, France, 2007, pp. 41–50.
[20] M. Petkovic, D. Prandi, N. Zannone, Purpose control: did you process the data for the intended purpose? in: Proceedings of the 8th VLDB International Conference on Secure Data Management, Seattle, WA, 2011.
[21] R. Pitofsky, et al., Privacy Online: Fair Information Practices in the Electronic Marketplace, a Report to Congress, 2000, Federal Trade Commission.
[22] M. Schunter, et al., The Enterprise Privacy Authorization Language (epal 1.1), W3C Working Group, 2003.
[23] V. Torra, Towards knowledge intensive data privacy, in: Proceedings of the 5th International Workshop on Data Privacy Management, and 3rd International Conference on Autonomous Spontaneous Security, Athens, Greece, 2010.
[24] X. Sun, H. Wang, J. Li, Y. Zhang, Satisfying privacy requirements before data anonymization, Comput. J. 55 (4) (April 2012) 422–437.
[25] L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (5) (2002) 571–588.
[26] H. Wang, J. Cao, Y. Zhang, Delegating revocations and authorizations in collaborative business environments. Special Issue on Collaborative Business Processes, Inf. Syst. Front. 24 (2008) 870–878.
[27] G. Wang, Q. Liu, J. Wu, Achieving fine-grained access control for secure data sharing on cloud servers, Wiley's Concurr. Comput.: Pract. Exp. 23 (12) (August 2011) 1443–1464.
[28] H. Wang, Y. Zhang, J. Cao, Effective collaboration with information sharing in virtual universities, IEEE Trans. Knowl. Data Eng. 21 (6) (June 2009) 840–853.