



ISSN: 2454-9940



**INTERNATIONAL JOURNAL OF APPLIED
SCIENCE ENGINEERING AND MANAGEMENT**

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org



www.ijasem.org

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

Mrs. K. Janani, Mr. S. Sugavanam, Mrs. K. Lavanya, Dr. J. Thilagavathi

Associate Professor ², Assistant Professor ^{1,3}

kjanani@actechnology.in, sugavanam.s@actechnology.in, klavanya@actechnology.in

Department of CS & BS, Arjun College of Technology, Thamaraiikulam, Coimbatore-Pollachi

Highway, Coimbatore, Tamilnadu-642 120

Abstract:

This article lays out algorithms for policy conflicting situations and suggests a methodology for purpose-based access management in distributed computing environments that aims to preserve privacy. Information that contains personally identifiable information is subject to the access policy, which this mechanism implements. The essential part is models for purpose-involving access control that express very complicated privacy-related regulations with many aspects. A subject's access rights to an object are determined by policy, which in turn is determined by attribute predicates, obligation actions, and system circumstances. When new access policies are created, there's a chance that they can clash with current policies. This might lead to policy conflicting difficulties. Confidential data cannot be adequately safeguarded due to policy disagreements. Efficient conflict-checking algorithms are devised and implemented after studying the structure of the related access control policy. The presentation concludes with a comparison of our study to comparable works, including EPAL.

1. Introduction

As consumers and businesses alike become more concerned about their personal information, privacy protection measures are taking front stage in modern advertising campaigns. Particularly for context-aware online services, this poses difficult concerns and issues about the use and security of private communications [6]. The question of who has permission to access private information and for what reasons is central to data privacy principles [2]. For instance, hospitals are only allowed to utilise patients' personal information for record keeping purposes; they cannot use it for advertising. Data gathering and access must have a purpose. The primary reasons for implementing a purpose-based approach are 1) the fact that customers have acknowledged that data usage differs from person to person and 2) the fact that basic privacy policies are concerned with the specific data objects used for specific purposes [20]. Technology in the information sector allows for the storage of many different kinds of user data needed for operational purposes. The vast majority of websites do in fact gather some kind of personally identifiable information from their visitors; for example, names, home addresses, email addresses, and postal addresses (Pitofsky, 97%). Many individuals feel their privacy violated since their personal information is gathered and utilised without their knowledge or agreement. The purpose-based approaches to data security are examined in this article.

The term "data privacy" is used to describe the rules that govern the disclosure of data and its use [1]. As an instance, a policy may provide that, in order to reveal the price of an airline ticket, it must be done so in conjunction with

"opted-in" clients, or that the agent will be required to reveal the price unless they have "opted-out" of it. There has been no research on access control techniques for implementing privacy rules, despite recent work on defining languages for doing so [22,11]. To better understand how to implement complicated privacy rules, Ni et al. [19] examined a role-based access control system that combines conditional privacy management with expressive condition languages. This system allows for variable interactions among permission assignments. The development of a formal approach to define and manage objectives, as well as the automated detection of potential conflicts between access regulations, are two examples of the many intriguing difficulties that persist. This is because "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," as stated by Al-Harbi and Osborn [4] and Adams and Sasse [3].

It is crucial to implement access control measures when sharing sensitive information over online services [14]. Traditional access models, particularly purpose-based access control systems, have not embraced the idea of privacy, despite the long-standing recognition of its relevance. In order to determine if access is necessary, a security officer must review privacy rules. Also, while creating new access controls to access sensitive data, administrators are human and may make errors [7]. Due to the many privacy requirements and the ongoing engagement of security officers, this method considerably raises the management burden in dispersed systems. This study provides a connection between models for access control and technologies that safeguard private information. Our first step is to examine the inconsistencies in access control regulations and construct a purpose-based access framework.

What follows is an outline of the rest of the paper: Our goals for writing this paper are laid forth in Section 2. In Section 3, we provide a purpose-based access framework that evaluates access controls and provides comprehensive information about their goals. In Section 4, we cover the structure and authorisation models of access control policies. We also show how creating a new policy might affect things via examples. Section 5 lays out the issues with access regulations and goals that might clash with one another, and it creates algorithms to identify such conflicts. Section 6 details the execution of the competing algorithms. Section 7 draws parallels between this paper's findings and those of similar works; these comparisons show how this paper's findings are significant. In Section 8, the article concludes and suggestions for further research are provided.

2. Motivations

Notable examples of distributed systems designed to accommodate privacy standards include the widely used P3P standard [27,11,13], where crucial procedures for private information take place. When it comes to relational database systems, Agrawal et al. [2] brought the idea of Hippocratic databases, which include privacy protection. An important part of their work is the use of privacy metadata, which includes privacy-authorizations tables for authorisations and privacy-policies tables for policies. Nonetheless, the ideas of purpose in relation to hierarchical structures, prohibition of purpose, and linkage of purpose with data pieces were not addressed. In order to implement privacy policies in database systems, LeFevre et al. [15] laid forth a method. Table semantics and query semantics were two types of cell level restricted disclosure enforcement that they developed; nevertheless, they neglected to address access control management. In order to

make data as useful as possible while keeping personal information as private as possible, Li et al. [16] developed generalisation boundary approaches. The authors analysed an access process management through a trust-based decision and ongoing access control policies, and they suggested a privacy-aware access control model for use in web service environments, drawing inspiration from the fact that the possible generalisation level leads to much finer level access control. But the idea of purpose was omitted. There was a lack of consideration for use access management and policy conflicts across purposes in the study of Ni et al. [19], who examined a role-based access model for purpose-based privacy protection. Advancements in access technology may pave the way for improved access management in the future by solving complex problems in areas such as modelling and design. In response to issues with privacy invasion, this article builds purpose-based access technology, which includes access control and complicated policy-structured models. Data disclosure to unaffiliated parties carries the risk of privacy infringement [2]. Once data is made public, the organisations that possess it lose control over it, and the data owners have no say over how it is used. To protect the confidentiality of publicly available information, the standard practice is to edit the data so that no one can associate it with a specific person [24]. Note that it may not be sufficient to anonymise the data if identifying information like names or social security numbers is still included in the leaked data. There are other instances when this kind of information was remained linked to the people it referred to after it was deleted from the disclosed data [23]. As a remedy to the challenge, Sweeney [25] suggested methods based on the idea of k-anonymity. Data mining is another setting where safe private information approaches like density-based clustering algorithms take place [18].

In today's world, data mining methods rock. Thus, data mining methods may enable recovery of sanitised data even after deleting private information from a database. Some of these methods alter or disturb the data in some manner; for instance, privacy-preserving association rule mining methods alter the data in a way that lowers the confidence of sensitive association rules [12]. An issue with these methods is the data quality that comes out of them; data that goes through too many changes could become useless [10].

Unlike privacy policies, which focus on the use of data objects for specific purposes, traditional access management systems track which users are doing what actions on which data objects [28]. As a result, traditional access management systems make it difficult to secure private information. Customers and data collectors often enter into privacy agreements that include things like "we use customer information for marketing purposes and to enable help us to resolve problems." For instance

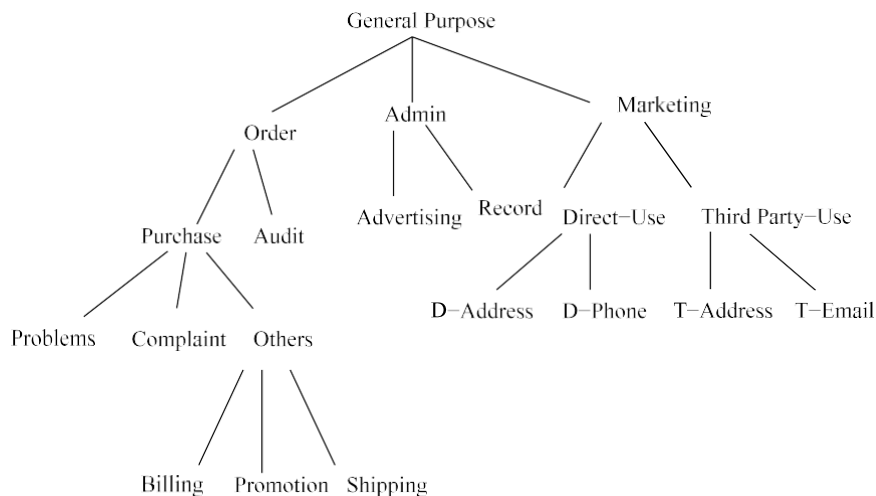


Fig. 1. Example of purpose structure.

in connection with services" that just mentions that customer data may be accessed for marketing and customer service reasons without identifying who exactly has access to the data. The difficulty in resolving policy conflicts is an additional obstacle to effective access control policies. For illustration's sake, let's pretend there are three different access control rules; if two of them don't clash with each other, then there may be problems when all three of them are applied at once. The purpose-based access management policy specification and enforcement utilising a rule-based language is the only subject of this study. We provide a thorough structure for managing purposes and data, with purposes organised hierarchically. Unlike conventional secure applications, which only use a single layer of protection, our method assigns many purposes to each data piece. Furthermore, the goals are hierarchical and subject to change. These specifications are more involved than those for standard secure applications [17]. This study examines the affiliation of a set of purposes with access control policies and the explicit restriction of purpose in order to provide adequate functionalities with the framework. We also create algorithms to identify and resolve conflicts, and we talk about the challenges with numerous access control rules. So far, no research has examined this kind of analysis for privacy-preserving purpose-based use management.

Purpose involved access control framework

3. A purpose-based access control framework (PACF) is developed in this section. Through the introduction of purpose-associated data models, expanded access control models, and intended and access purposes, PACF bolsters purpose hierarchy. In order to determine the purpose of access in database systems, authorisation techniques from access control models are intended to be used.

Purpose A purpose statement explains why data is being collected and why it is being accessed [19]. A Purpose Tree (PT) is a hierarchical relational database that organises a collection of purposes (P) in a tree form. In this model, each node represents a purpose in P and each edge represents a specialised or generalised link between two purposes. Figure 1 shows a purpose tree in action.

Consider a purpose tree with two nodes, P_i and P_j . If there is a downward route in the tree from P_i to P_j , then P_i is senior to P_j , and vice versa. There are partial links between purposes based on the tree structure of purposes. Assume that P is a collection of purposes in PT and that PT is a

purpose tree. As a purpose belonging to the set P , P_u 's senior purposes are

The set of all nodes that are senior to P_u is indicated by $\text{Senior}(P_u)$. Take $\text{Senior}(\text{Record})$ as an example: it's equal to $\{\text{Admin}, \text{General Purpose}\}$.

in Figure 1. All nodes that are junior to P_u make up the set $\text{Junior}(P_u)$, which represents P_u 's junior purposes. Take, for example,

The junior property of the admin property is $\{\text{Advertise}, \text{Record}\}$.

Next, we go into the specifics of the model-based access purpose authorisation and verification, after which we construct an access control model by including policy language and purposes. It stands to reason that a data piece should be accessible if the stated or implied goals for the data in the privacy rules support the intended use of the data. Users are given access authorisations depending on the data's purpose, as well as their duties and constraints. Prior research has presented authorisation models for access control, such as the pre-Authorizations model and the ongoing-Authorizations model [26], and this study examines the role of authorisations for access purposes in access control policies.

4. Access control policies

Following an overview of the fundamental ideas behind goals, we provide the framework for an access control policy [8]. For this system, certain policies have been outlined. Consider a typical computer system that has resources or data that must be

shielded from prying eyes. To ensure user privacy, access control policies are built with authorisation models and policy operations in mind.

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

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

A person or organisation making a request to access the resources is identified by the words of the topic. Any change (like removing a file) made to an access resource is considered an action. The phrase "resources" designates a selection of items, often including sensitive information, to which only authorised individuals have access. An action's purpose is the reason (or combination of reasons) why a subject plans to carry it out. "Obligations" are rules that the subject must obey in order to get resources, and the condition is a Boolean statement (a predicate). For example, in order to use Skype, users are required to approve the agreement of privacy policy during the installation process. Since there isn't enough room to cover all of the possible situations in this article, we won't. When it comes to conventional access control rules, the ideas of subjects, actions, and resources are interchangeable. Achieving fine-grained policies is the goal of purposes. With no unintended consequences, the purpose checks for context attributes. If there is a side effect, we must take into account other considerations, such as the authentication process's criteria and duties. We touch on responsibilities briefly in this article, but we don't go into depth on them. In the previous part, we

established that private access rules cannot exist without a goal, which is the driving force behind resource collection.

Two instances of authorisations, one positive and one negative, are given below. There are two regulations in the sample security policy.

During business hours, Hua may access purchase information for marketing purposes. Whenever Christine wants to change phone numbers for record purposes, she is unable to do so.

S stands for Hua, A for read, R for buy information, P for marketing, and C for 8:00 am to 6:00 pm in the first rule. In the cases, there are no demands. S stands for Christine, A for update, R for phone number, and P for record in the second example with negative authorisation.

$C = anytime.$

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 $Pu = (AIP, PIP)$ 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 $PIP = \emptyset$, 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, \emptyset)

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

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', \emptyset).

Both of these rules allow Tony to access email addresses, although for different reasons than P2. When put together, what are the outcomes of these two policies? Our standard procedure calls for Tony to use P2 to get access to the email address for purchase purposes and P5 to gain access to the email address for complaint purposes.

Both rules exist, but one is for purchases and the other is for complaints. That's where the similarities and differences lie. What checks will the system do? In order to access email addresses with consent requirements, should the system validate the complaint? *PAC* does this by seeing various access regulations as being connected in a conjunctive fashion.

In other words, all of U's access policies pertaining to $((d, a), Pu)$ need to be evaluated if user U wishes to access right a on data d for purpose Pu. There must be a policy for you to read the material, and you must be able to fulfil all of the policies' requirements. The purpose of implementing a new access policy is not to loosen access restrictions but to tighten them when it pertains to the same user, same data, same rights, and similar circumstances as existing private rules. Privacy officers don't need to draft brand-new access rules if they want to soften the access environments; they may just update the ones that are already in place.

Could we just replace the two existing private access policies in *PAC*, denoted as $(u1, (r1, d1), pu1 \wedge pu2, c1, \emptyset)$, with a new one, denoted as $(u1, (d1, r1), pu2, c1, \emptyset)$? Here is our policy on P2 and P5:

Tony is involved in the following process: P6: (EmailAdd, Read), Complaint \leftarrow Purchase, OwnerConsent = 'Yes', χ).

Since Complaint is subordinate to purpose Purchase in the purpose hierarchy structure shown in Figure 1, the union of Complaint and Purchase is Complaint. We get "Tony can read customers' Email address for Complaint purpose if the customers agree to do so" when we translate P6 into plain English. Something is lost in the translation, thus it is not accurate. Tony is unable to obtain email addresses for the sake of troubleshooting problems and other unrelated purchases. The P5 context variable "purchasing purpose" is to blame for this. The variables "purpose of purchase," "complaint," and "others not included" categorise the order values into three distinct groups. So, all three types of consumers are covered by P2, but only email addresses used for complaints are covered by P5. All save the Complaint reason for access email addresses are eliminated when P2 and P5 are combined.

To do this analysis, you need to understand the concept of dividing context variables [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.

Since the joint sets of Advertising and Record, D-Address, and D-Phone are not empty, according to the SCV definition, Direct-Use and Admin are not SCV, but Order is. An important part of the paper's study is the concept of SCV. We can now respond to the issue that was asked earlier: both pu_1 and pu_2 may be safely rewritten as $pu_1 \wedge pu_2$ if they do not include SCV or if the values of the SCV they do involve are the same.

Take a look at these two access policies:

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

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

P7 and P8 can be revised as:

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

Similarly, the following two access policies:

P10: (Tony, (EmailAdd, Read), Shipping, OwnerAge ≤ 13 , \varnothing)

P11: (Tony, (EmailAdd, Read), Record, OwnerAge ≤ 13 \varnothing)

P12: (Tony, (EmailAdd, Read), Shipping \wedge Record, OwnerAge ≤ 13 , \varnothing)

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', \varnothing)

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

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

Here we'll go over some of the many PAC model instances when policies are at odds with one another. Complying with intricate security and privacy regulations is a significant challenge, particularly for big businesses. It is more likely that a security policy will include portions that are inconsistent and contradictory if it is more complicated.

Think about these rules:

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

The problem-solving session will take place from 5 PM to 11 PM and will be led by Christine.

Since P19 and P20 provide distinct functions, they are not incompatible with one another. These two policies use the SCV Order for distinct reasons and assign different values to it. An incomparable policy is one that serves two distinct but related goals; specifically, one that employs a SCV with two separate but complementary sets of values.

Definition 5.1 Assume that two access control policies have two goals, pu_i and pu_j . If there is a common SCV with different value sets in purposes pu_i and pu_j , we argue that the two purposes are incomparable. In every other case, we describe pu_i and pu_j as similar objectives, denoted as $pu_i \approx pu_j$.

Two permission assignments with similar goals are as follows: In P21, the variables read and order information are associated with the purchase and the time period from 9 AM to 5 PM.

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

A data request that happens between 9AM and 5PM with a billing purpose might be authorised since P21 enables data access during that time for purchase purposes and P22 allows data access during the same period for billing purposes. There is no vacant space at the intersection of the value sets of the context variable Order in the two access rules, so they are compatible with one other and serve complementary objectives.

Our goals could be complementary or even contradictory.

In P23, the variables read and order information are associated with the purchase that takes place between 5 PM and 11 PM. Christine is auditing the process from 5PM to 11PM.

P24 grants partners' access with Audit from 5PM to 11PM, whereas P23 stipulates that Christine may view Purchase order information within that time. Therefore, it is not possible to have both purchase and audit purposes for data access requests. Consequently, these two access rules will prevent any data request from being approved. Because they serve different goals, these two authorisations are incompatible; in other words, no single value for the context variable Order could satisfy both of these requirements.

Definition 5.2. Consider two access policies with similar goals, pu_i and pu_j . If there is at least one shared context variable between pu_i and pu_j with different value sets, denoted as $pu_i \not\approx pu_j$, we say that the two variables are conflicting purposes. If not, we state that the aims of pu_i and pu_j are compatible.

Various access rules may impose competing demands; take them into consideration:

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

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

P25 and P26 are at odds with each other because, after authorising a data request, the system is unsure of which obligation—Notify or Notify with Opt-out—should be implemented.

Two responsibilities oi and oj are in conflict with one other, and we indicate this as $oi \text{ }^3\text{ } oj$.

Our explanation of competing access regulations is based on the concepts and examples provided before.

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

$$(ui = uj) \wedge (ri = rj) \wedge (di = dj) \wedge (ci = cj) \wedge (pui \text{ }^3\text{ } puj) \\ (ui = uj) \wedge (ri = rj) \wedge (di = dj) \wedge (ci = cj) \wedge (pui \approx puj) \wedge (oi \text{ }^3\text{ } oj)$$

To avoid confusion while implementing access policies, PAC should identify conflicting rules and eliminate one of them.

Identifying systems

To ensure that access control rules are consistent, it is critical to identify policies that conflict with one another. Algorithms for detecting purpose-conflicts and access-control policy conflicts are presented in this section. A disjoint test for the value sets for a variable in the different conditions is performed after sorting context variables used in conditions according to their name. This is the main point of the method.

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: $pu1$: Sort context variables used in $pu1$ according to their name

2: $pu2$: Sort context variables used in $pu2$ according to their name

3: for(integer $i = 1$ to $|pu1|$)

4: { for(integer $j = 1$ to $|pu2|$)

5: { if $pu1[i].name = pu2[j].name$ //Common context variable

6: then

7: { if $pu1[i].SCV = True$ // $pu1[i]$ is an SCV

8: {if disjointTest($pu1[i].value$, $pu2[j].value$) = 'False' $pu1[i].value$ and $pu2[j].value$ have joint value sets, no conflicts between $pu1[i]$ and $pu2[j]$

9: then $j++$ //check the next purpose in $pu2$

10: else

11: Return True //Conflict purposes }

12: else $j++$ //check the next purpose in $pu2$ }

13: else $j++$ }

14: $i++$ //check the next purpose in $pu1$

15: Return result

What follows is a method for detecting access control policies, based on the Purpose-Conflict algorithm. In order to determine if policies are conflicting, the algorithm first checks for purpose conflicts. Otherwise, you should look at the responsibilities to see whether there is a contradiction between the policies.

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  $\{(po_1.o \cap po_2.o) = \emptyset$  //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

Here we show how the algorithms for the access control policy were implemented using Microsoft Visual Studio. The data structure, including characteristics, in each access control policy is not a concern when we use Microsoft Visual Studio technology. If you ever need to update or establish another access policy, you may easily do so by adding characteristics to the current policy database. Built on the Windows platform using the XAMPP environment

(<http://www.apachefriends.org/en/xampp.html>), this web-based project makes use of the MySQL database and the Apache web server. Subjects, actions, resources, intents, and responsibilities data is stored in the open-source MySQL database because of its high dependability, simplicity of use, and high performance during policy implementation. We keep track of the organisational framework, including policies, purposes, resources, and more. Apache has come a long way from being the first practical alternative to Netscape Communications Corporation's web server; it now outperforms all other web servers in terms of features and speed, and it makes it easy to migrate apps to any OS that uses Google Chrome. It can reach new heights using HTML5 with the help of Microsoft Visual Studio.

There are a lot of moving parts in putting the algorithms for access control policy into action, such as:

- 1) The database's structure, which includes its purpose, resources, and policies for controlling access
- 2) Divergent goals, responsibilities, and policies
- 3) Policy requirements and duties.

Our clients must use up-to-date web browsers that support cookies, such as Chrome, Firefox, Internet Explorer 6 or later, or Mozilla Firefox. In order for the computer to access the system, it needs an Internet connection.

Designing Databases

Many tables, including Policy, Resource, and Purpose, make up the database used to implement the PAC paradigm. Take the PAC model, a tuple of the form (Subject, Action, Resources, Purpose, Condition, Obligation), as an example. It defines the policy.mdf table. The Policy Table is defined in Fig. 2 below.

The Interface for the User

Our goal in designing this graphical user interface was to streamline the process of adding and deleting rules during deployment. If you look at Figure 3, you can see the GUI. Instead of manually entering the function call mentioned before, this interface was built using Microsoft Visual Studio 2010 Ultimate. To trigger the generation of access control policies, the MySQL database is used. The implementation is designed to be user-friendly for administrators, who just need to outline the hierarchy of purposes and the structure of obligations.

Figure 3 displays a data grid displaying all database policies. You may edit and remove them to make changes. A policy ID must be entered in the text box provided on the top of the data field before the Submit button can be used to access the resource. Click the create Policy button to create a new policy, as shown below.


	Column Name	Data Type	Allow Nulls
	PolicyId	bigint	<input type="checkbox"/>
	Subject	varchar(50)	<input type="checkbox"/>
	Action	varchar(50)	<input type="checkbox"/>
	Resource	varchar(50)	<input type="checkbox"/>
	Purpose	varchar(50)	<input type="checkbox"/>
	Obligation	varchar(50)	<input type="checkbox"/>

Fig. 2. Policy structure.

Enter policy Id:

	PolicyId	Subject	Action	Resource	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

Fig. 3. Creating policy.

Comparable Policy

Subject:

Action:

Resource:

Purpose:

Obligation:

Fig. 4. Comparable policy.

Similar strategies

Fig. 4 shows that the current policy 16 and the newly generated policy with the following data are similar; both have the same Subject, Action, Resource, and Obligation, but the new policy's goal is Billing instead of Purchase. The new policy will be added to the database and a notice about similar policies will be shown because of the comparable attribute.

Potential for conflict of interest

The following scenario plays out whenever a new policy with the data shown below is planned to be added to the database: we compare it to policy 16, which also has the same Subject, Action, Resource, and Obligation, but its purpose is Audit rather than Purchase. As a result, we get a message about a conflict of purpose, and the new policy cannot be added to the database (Fig. 5).

Duty incompatibility

Concerning the implementation's handling of duty conflicts, it's quite comparable to the conflict of objectives. This policy is compared to an existing one, number 16, which has the same Subject, Action, Resource, and Purpose but an Obligation of NAT instead of NA. The data provided below is from the latter policy. Figure 6 shows that they are unable to add to the database and will instead send us a warning about a conflict of responsibility.

First, administrators will have an easier time managing access control policies with different objectives and responsibilities thanks to the implementation's user interface, which allows them to add and remove policies from the database without technical knowledge.

Conflict of Purpose

Subject:	<input type="text" value="Christine"/>
Action:	<input type="button" value="r"/>
Resource:	<input type="button" value="OrderInformation"/>
Purpose:	<input type="button" value="Audit"/>
Obligation:	<input type="text" value="NA"/>
<input type="button" value="Confirm"/>	

Fig. 5. conflict of purposes.

Conflict of Obligation

Subject:	<input type="text" value="Christine"/>
Action:	<input type="button" value="r"/>
Resource:	<input type="button" value="OrderInformation"/>
Purpose:	<input type="button" value="Billing"/>
Obligation:	<input type="text" value="NAT"/>
<input type="button" value="Confirm"/>	

Fig. 6. Creating policy with conflicting obligation.

assistance that is useful for the organisers of the system; 2) the system resolves the issue of competing policies; it accompanies conflicts of purpose as well as duties. 3) Using Microsoft Visual Studio, a lifetime of platform and technology portability made simple, for implementation. We plan to address generalised time restrictions and the policy-creation workflow in future work, but for now we will only state that we will do so.

7. Comparisons

Here, we provide a concise review of the relevant literature and contrast the PAC model with other relevant studies. A conditional role-involved purpose-based access control paradigm [14], privacy-aware role-based access control [19], and the enterprise privacy authorisation language (EPAL) [22]

are all studies that are closely linked to this study. To fully allow the expression of very complicated privacy-related regulations, including elements like objectives and duties, Ni et al. [19] presented a family of models that expand the famous RBAC model. Among these models are the Core, Hierarchical, Conditional, and Universal P-RBAC frameworks. There are three ways in which their work differs from ours. To begin, the conditions and linkages in role-based access control are the main emphasis of their work. In contrast, our research has focused on the use access control model's purpose hierarchy structure inside access control rules. Secondly, the criteria are the root cause of the discrepancies between the two P-RBAC permission assignments that are addressed in their study. Neither the structure of access purposes nor the effects of introducing a new policy with diverse goals are examined. Our research, on the other hand, has focused on the hierarchical structure of purposes and how new access control measures affect them, paying particular attention to the issue of competing three purposes.

For the purpose of regulating data handling activities in IT systems in accordance with fine-grained positive and negative authorisation rights, EPAL [22] is a formal language for creating corporate privacy policies. All deployment specifics, including data models and user authentication, are abstracted away so that the focus may be on basic privacy authorisation. An EPAL policy lays forth the rules for what users must do in order to protect personal information, as well as the hierarchies of data types, user categories, and purposes. "Purposes" provide the model of the data's intended use (such as processing a reimbursement for travel expenses or conducting an audit). Here are some key distinctions between PAC and EPAL. First, PAC's ability to consolidate the enforcement of privacy policies and access control policies into a single model is a key component of its architecture. In contrast, no access control paradigm was considered when developing EPAL. Secondly, EPAL did not address the conflicting policies issue, which means there is a gap in responding to data access requests [5]. However, PAC has conflict detection features to ensure that no conflicts occur during policy generation procedures, which keeps private information safe. Third, PAC takes its core concepts from EPAL; although EPAL's purposes only state why data is being collected, without delving into other issues, such as privacy concerns, PAC's purposes include extensive analysis and conflict algorithms.

A multi-purpose privacy-preserving access control was suggested in the article [14]. The concept incorporates conditional purpose in addition to approved and banned purpose. Using dynamic roles, we develop and study the conditional purpose-based access control model's structure. A compliance-achieving algorithm is created.

the dynamic use of Role-based access control (RBAC) to demonstrate the calculation between access purposes and intended purposes, which enables conditional purpose-based access control. Nevertheless, the article focused on expanding traditional access control models to include additional privacy-preserving measures in data mining environments, rather than analysing the structure of access control policies, related models, purposes, obligations, and conflicts among access control policies and purposes.

8. Conclusions and future work

In this work, we have covered the topic of distributed computing systems' purpose-based access control rules, together with their restrictions and requirements. Not only have we researched the framework for access control, but we have also examined the components of access policies, such as resources, topics, actions, and duties. We have also considered the potential effects of new policies and the disputes that may arise as a result of them. The development of algorithms has facilitated the ability of systems to identify and resolve issues. In addition, the algorithms' efficiency and applicability are shown by the experimental results. Several areas of prior work have been considerably expanded upon in this study. For instance, the following: purpose-involving access control; access control policies; and the generation of a new access policy free of conflicts. There is a lot of ground to cover in the early stages of research on purpose-driven access control rules. Duplicate access policies may be included in PAC. As an example, when it comes to P8, P7 is unnecessary. Possible directions for our future efforts include formalising the redundancy and finding remedies to it.

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.