A Mathematic Model for Securing Private Data in an Outsourcing Environment

Yuan Zhang, Richard J. Enbody, John R. Lloyd

Michigan State University, USA

Abstract

Outsourcing, as a popular business model, helps enterprises cut operating costs, but at the same time it also puts the enterprise's sensitive data in danger. How to ensure that private data is safe when it is distributed across the enterprises along an outsourcing chain is a critical security issue. Based on the reality of wide usage of workflow in enterprises and the observation that the workflows specify the relationships and dependences as well as the flow of data among tasks/enterprises, we propose a framework, which is built on top of workflow management systems, to protect private data by executing the “right” workflow at runtime. Our mathematical model discovers the use purpose of a workflow instance and ensures the preservation of intent purpose of data across different enterprises, so that the data privacy is guaranteed along the outsourcing chain.

Keywords: Privacy Protection, Outsourcing, Workflow, Information Flow, Access Control, Purpose

1 Introduction

Dear UCSSF Medical Center,

Your patient records are out in the open... so you better track that person and make him pay my dues.

A woman in Pakistan indirectly working for the University of California at San Francisco (UCSF) Medical Center sent this email which threatened to post patients' confidential files on the Internet unless she was paid. To show she was serious, actual patients’ records were attached (Lazarus 2003). This email sent shock waves through the US medical establishment and changed California's privacy laws.

The story behind this email is outsourcing, an increasingly popular business model. In the UCSF Medical Center the work of transcribing the oral notes, which are used to record the patients' discharge information, was outsourced to Transcription Stat, a California-based company. That company in turn outsourced the transcription work to Medical Data, a Florida-based company. The outsourcing was supposed to end here, but it did not. The Florida company further outsourced work to a Texas company, Tutranscribe, which eventually outsourced work to a woman in Pakistan. Later, the Texas company failed to pay the woman, which prompted the email sent to the UCSF Medical Center.

From this example we learn that 1) as the work was outsourced, work-related information, including private data of the customers, had to be shared across the companies along an
outsourcing chain; 2) once the information was released into the outsourcing chain, the owner's control on the information was lost.

Privacy protection is a major concern with respect to information protection in enterprises. Several projects exist to protect privacy and three, P3P (W3C 2002), E-P3P (Karjoth 2002) and PBAC (Byun 2005), are most useful with respect to enterprise privacy. However, these approaches focus on protecting privacy inside an enterprise, which helps the enterprise to keep the promise made to the customers, namely the owners of the personal information. We need a new solution to protect privacy across enterprises along an outsourcing chain.

Given that the intent of the information owner (e.g. the customer) is known, briefly our work will
1. determine the intent of the users (e.g. the employees of the enterprise) who need the data (use–purpose discovery)
2. select workflows that will satisfy the intent of the owner even as data is passed across enterprises (intent-purpose-oriented sub-workflow selection)
3. ensure that the business processes within enterprises maintain the intent of the owner (intent-purpose-controlled self-adjustment of workflow execution)

Within our framework the UCSF Medical Center data would not have been released.

This paper is organized as follows: background knowledge is given in section 2, related work is introduced in section 3; section 4 describes our framework in detail, the formal model and use purpose discovering algorithm are introduced in section 5, and we conclude our paper in section 6.

2 Background

In this section we introduce some basic concepts to help readers understand our framework.

2.1 Workflow and Enterprise

Daily business processes are the core components of modern enterprises. In order to control the execution of the business processes, Workflow Management Systems (WFMSs) are widely used. In such systems, the business processes are modeled as workflows. WFMSs provide environments for creating, executing, monitoring and administrating the workflows. A workflow is defined as a set of coordinated activities that achieve a common business objective (Hollingsworth 1994). An activity, also called as a task, is a logical step or description of a piece of work that contributes toward the accomplishment of a business process (Georgakopoulos 1995). A workflow not only contains the task's relations and dependences that reflect the coordinated activities of a business process, but also indicates the flow routine of data that are processed by the business process. The customers' private data are a part of the data processed through the workflows. Workflows indicate how the data is used by the activities/tasks in a workflow management environment, so we can say that workflows distribute data. If data is released to a safe workflow, the security of the data will be assured. If the data is released to an unsafe workflow, data leakage is possible, maybe likely. In other words, if we can control the workflows, we can control the (re)distribution of the data passed through the workflows. We observe that integrating the privacy protection mechanisms with workflows provides a more efficient way to help the enterprises to protect customer privacy, especially in outsourcing environments.
2.2 Process Modeling: Process Grammar

In this paper Process Grammar (Baldwin 1995) is used to specify workflows. Process Grammar represents a workflow as a flow graph, which consists of nodes and directed edges. Nodes have two types: task (oval) and data specification (rectangle). Tasks have two sub-types, logical task (divisible, single-oval) and atomic task (indivisible, double-oval). The directed edges connect tasks and data specifications, which indicate the generating or consuming relationship between the tasks and the data. Process Grammar does not only explicitly represent the control flow of (logical or atomic) tasks, but more importantly for us it also clearly shows the data flow. The usage of logical task has several advantages. First, logical tasks simplify the representation of complicated workflows by organizing them into hierarchical structures. Second, the logical tasks allow dynamic workflow instance generation, which provides flexibility to the workflow management systems.

Figure 1. (A) A workflow instance, (B) The hierarchical workflow used to generate the instance

A workflow example is shown in Fig.1. A workflow instance (Fig 1.A) is a running workflow for a particular business process. And this instance is dynamically generated by a set of sub-workflows (Fig 1.B). How to select a suitable sub-workflow for a logical task is beyond the scope of this paper. In this paper we only focus on the security impact of sub-workflow selection. More detail is provided in the framework section.

2.3 Privacy protection and Purpose

Private information is collected either from individual customers or from other partner enterprises for certain predefined reasons, and will be processed according to certain predefined guidelines. The reasons and guidelines are specified in the form of enterprise privacy statements which are shown to the customers and other enterprises. How to guarantee that private information is used for the predefined reasons and along the predefined guidelines are the questions that have to be handled by the enterprise security systems.

Conventional access control systems, such as Mandatory Access Control (Sandhu 1994), Discretionary Access Control (Sandhu 1994) and Role-based Access Control (Sandhu 1994), do not support privacy protection directly. Those systems focus on answering an authorization question: “Who can perform what action on which data?” and control accesses in terms of Subject, Action and Object, which is insufficient to handle accesses to private data. In order to support the additional requirements from privacy protection, one more entities, Purpose, is added to the access control system (Byun 2005, Park 2004).
Purpose describes “Why and How”, namely the reasons and guidelines, for which the private information is to be used. In our workflow-based framework we define three types of purpose: intent purpose, designed purpose and use purpose. The formal definitions of the three types of purposes will be given in Section 5.1.

An intent purpose, from the information owner's perspective, defines the reasons and/or guidelines for which the information is to be used. This is a mandatory specification, which must be complied by a user who handles the data. Intent purposes can either be defined by an information owner or proposed by a collector and approved by the owner when the information is collected. The intent purpose, held by an information keeper (the owner or someone who currently has the information), acts as a criterion that needs to be satisfied before the information is released to a receiver (someone who needs the data). The intent purposes are predefined and static. An intent purpose has two different formats: Allowed-Intent Purpose and Prohibited-Intent Purpose. By using both types, especially the prohibited-intent purpose, it is easier to specify a fine-grained access policy.

A designed purpose, from the workflow designer's perspective, specifies how and for what the input information will be processed by the workflow. Because a single workflow could be reused, based on the context in which the workflow is used, the actual purpose of the workflow may different from the designer's perspective. A designed purpose acts as a guideline, when other dynamic detail of the workflow instance is not available.

A use purpose, from the information-user perspective, describes the actual reasons for what the information is required and the actual guidelines how the information is to be processed. A use purpose is held by a workflow instance, and is implied by the actual business activities that are performed by the users/employees. Generally speaking, the designed purpose and the use purpose of a workflow instance are the same. However, a workflow instance is a dynamically generated combination of the set of sub-workflows, which may result differences between its use purposes and designed purposes. Use purposes need to be provided to the information keeper when information is required. The goal of privacy protection is to ensure that when the information is released from a keeper to a receiver, the use purposes of the workflow instance must comply with the intent purposes of the data.

Similar terminology of purpose has been used in Byun's work: a Purpose-based Access Control Model for Database (Byun 2005). In their work, two types of purposes are defined: (1) intended purpose, which is predefined and is associated with data in a database, and (2) access purpose, which is determined based on the role attribute of a given user. Their terms are similar to our intent purpose and use purpose respectively. Our use purpose is discovered from an instance of a workflow and a workflow hierarchy (details are given in section 5.4) instead of roles of users. The terminology that we use is more suitable for the context of our framework. For clarity: use purpose is more accurate to represent the meaning of “why and how the data IS TO BE used.”

3 Related work

In order to automatically protect the private information stored in an IT system, representing the enterprise privacy statements in a standard, machine-readable format is necessary. The Platform for Privacy Preferences (P3P) is a standard of W3C(2002) for a language to specify privacy
policies. A P3P policy is an XML document that describes data collection practices. It provides a base schema for the data collected and a vocabulary to express the purpose, the recipients, and the retention policy. P3P captures common elements of privacy policies, but it does not provide technical mechanisms to check a given access request against the stated privacy policy. This is the value added by this research.

Bonatti et al. (2001) argued that use-based restrictions are critical for privacy protection. Their language characterized a data user as the triple: \( \text{(user, project, purpose)} \). Projects are named activities registered at the server for which different users can be subscribed, and which may have one or more purposes. Each user and project is associated with a profile, which captures properties such as name and title.

On the other hand, Fischer-Hubner (2001) developed a task-based access control model that has the notion of purpose. In their model, data can be accessed in a controlled manner only by executing a task. The task's purpose must comply with the purposes for which the data was collected or there has to be consent by the data subjects. However, this model only focuses on single simple tasks rather than complicated business processes. In their model these tasks are clearly predefined and fixed, which is not flexible enough to handle the dynamically changed business processes.

The Platform for Enterprise Privacy Practices (E-P3P) (Karjoth 2002) defines a technology for policy enforcement within an enterprise and provides a language to express privacy-specific restrictions. Their \textit{sticky policy paradigm} says that data and corresponding policies should be kept together, and when the data is required, the policies should be examined. In their language the policy is specified in the form of \( \text{(purpose, data, user, operation, condition, obligation)} \). Their approach is further refined and described in the Enterprise Privacy Authorization Language (EPAL) specification (IBM 2003). The eXtensible Access Control Markup Language (XACML), an OASIS standard (OASIS 2005), is a similar language used to represent access control policy. In XACML v2.0 the element \textit{purpose} is added to support policies for privacy protection.

The proposals mentioned above mainly focus on protecting privacy within one enterprise. These systems help enterprises keep their privacy protection promises made by the enterprises to their customers. However that is not enough for privacy protection in an outsourcing environment. In such an environment, we need a multi-level systematic solution, because there are multiple enterprises involved. We believe that a comprehensive security solution requires at least two levels. One is the protection system between enterprises; the other is the protection system within each enterprise. The goals of the later systems have been addressed in (W3C 2002, Karjoth 2002, Bonatti 2001, Fischer-Hubner 2001). In this paper we focus on the protection system between enterprises. The big difference between our system with the systems for inside enterprises is that we emphasize the purpose for information and the routines along which the information is used rather than focusing on the users who handle the data.

4. Framework
In this section we will describe an access control framework built on top of workflow (represented in terms of Process Grammar) for data privacy protection. Purpose is the critical entity in our framework. The support of purpose consists of three parts:

1. determining the intent of the users (e.g. the employees of the enterprise) who need the data (use–purpose discovery)
2. selecting workflows that will satisfy the intent of the data owner (intent-purpose-oriented sub-workflow selection)
3. ensuring that the instances of business process maintain the intent of the owner (intent-purpose-controlled self-adjustment of workflow execution)

### a. Use-Purpose Discovering

In existing frameworks (such as E-P3P (Karjoth 2002) and PBAC (Byun 2005)), purposes (in a hierarchy) are defined by the Privacy Officer in a static way. The same purpose hierarchy serves both for information collecting (intent purpose) and information processing (use purpose). However, in the real business world the intent purposes and the use purposes may not be the same as we expect. Our work addresses this problem.

![Figure 2. Pre-defined Static Intent-Purpose Hierarchy.](image)

Suppose we have a set of intent purposes, which are organized into a static hierarchy (fig2), for the personal information (e.g. name and address) of a patient in a hospital. A medical test will be ordered for the patient and the identification of the test results is the issue. Here, we have three purposes: General-Purpose, Order Test and Sharing. From the hierarchy we know that General-Purpose implies Order Test and Sharing, and there is no relationship between Order Test and Sharing.

![Figure 3. A patient orders a sample test from a hospital and provides his name and address. His personal information is shared with lab2, but not with lab1.](image)
Figure 3 shows a workflow that is used to handle the procedure of a sample test. From the workflow we know that the test consists of two parts that are processed by Lab1 and Lab2 respectively. The test result from Lab1 is sent back to the hospital and then forwarded to the patient by the hospital. In such a case, the name and address of the patient are not shared between the hospital and Lab1. From the point of view of the patient, his personal information (e.g. name and address) is only used for “order test”. On other hand, the test result from Lab2 is sent to the patient directly by Lab2. In such a case, the name and address of the patient have to be shared between the hospital and Lab2. From the viewpoint of the patient, his personal information is used for “order test” and “shared” to the third party outside the hospital. By analyzing the workflow we can discover the use-purpose hierarchy (fig4) of patient's personal information. And the use-purpose hierarchy may different from the intent-purpose hierarchy.

![Use-Purpose hierarchy discovered from the workflow.](image)

This example shows that the use-purpose of the data is decided by the workflow that processes the data. Analyzing workflows before releasing data to them will help prevent data misuse. By scanning the data specifications and the tasks which take the data specification as inputs, we can discover the use-purpose of the data for the workflow. The formal method of discovering the use-purpose is given in section 5.4. That formal algorithm is the core of our framework. The remaining parts support it: (1) intent-purpose-oriented sub-workflow selection and (2) intent-purpose-oriented self-adjusting of workflow execution are built on the top of this algorithm.

### b. Intent-Purpose-Oriented Sub-Workflow Selection

In the previous example the hospital acts as an information receiver and it has the capability to select a suitable workflow to implement “order test” that complies with the patient's choice: either allowing “sharing” or not. In this section, the hospital becomes an information sender, and sends information to its subcontractors (Figure 5).
In this example, before the hospital subcontracts the “sample test” task to Lab1, it asks Lab1 to send the workflow instances used to perform the “sample test”. As shown in figure 5, Lab1 has two different approaches: fig5(A) and fig5(B). The hospital uses the same method described early to discover the use-purpose hierarchies of these two instances that can be used by Lab1. According to the patient-specified intent-purpose, the hospital can require Lab1 to select workflow (A) or (B). If Lab1 is operated by the hospital and Lab2 is not, and the patient does not want to share his personal information other than the hospital, in such a case, the hospital will require Lab1 to use workflow (A). This scenario is called intent-purpose-oriented sub-workflow selection.

Intent-purpose-oriented sub-workflow selection works in two different ways. The first approach is one-level-ahead use-purpose discovering. In this approach the system only discovers the use purpose of the sub-workflows that are immediately used to handle the data. This approach provides the most flexibility with respect to privacy protection. This approach is especially suitable for an outsourcing environment because the detailed implementations in the subcontracted companies are invisible to the current company in most cases.

The second approach is to have the workflow management system recursively discover all use purposes down to the lowest level sub-workflows that only contain atomic tasks. The advantage of this approach is that once all use purposes are discovered and have been determined to satisfy the customer's intent purpose, the privacy protection is guaranteed. The disadvantage is that this approach is only suitable for a static workflow instance, and any change of the instance in the future could result in information leakage.

\section{Intent-Purpose-Oriented Self-adjusting of Workflow Execution}
In order to generate the intent purposes one of the methods uses an opt-in and opt-out purpose list (Bonatti 2001) which is usually provided by a company (e.g. the hospital). A patient can select or de-select purposes from the list to create his own set of intent purposes for his data. If the list is big, the possible combinations of the intent purposes will be huge. It is difficult to create and maintain a huge number of workflows and sub-workflows to satisfy every possible intent-purpose combination. The workflow management system can start with a complete workflow for all purposes (Fig6) when the input data is ready. At run-time the system can automatically detect the use purpose of the next task. If the use purpose complies with the intent purposes, the task will be executed as scheduled, otherwise the execution of the task will be blocked. Suppose the patient gives his information for "sample test" only. After the whole workflow is loaded into WfMS, the system should automatically disable two branches as shown in figure 6.

5. Mathematic Model

In this section we extend our framework (Zhang 2006) by providing a model to formally describe the key components of our framework and also provide an algorithm to discover the use purpose of a workflow instance before we release the data to it in an outsourcing environment.

a. Basic Concept

The concepts used in this model are formally defined as follow.

Definition 1. A purpose describes reasons and/or guidelines for which the data is to be used. We denote $p$ as a single purpose, and $P$ as the global set of purposes. $p \in P$.

Definition 2. The purpose hierarchy, $PH \subseteq P \times P$, is a partial order relationship on $P$, denoted by $\leq_p$, also called the “purpose dominance”. Given $p, p' \in P$, $p \leq_p p'$ holds if $p$ precedes $p'$ in the order, graphically represented by a directed edge going from $p$ to $p'$ in the hierarchy. In other words, the purpose $p$ implies the purpose $p'$.

Rule 1. (Transitivity) Given $p_1, p_2, p_3 \in P$, if $(p_1 \leq_p p_2)\in PH$ and $(p_2 \leq_p p_3)\in PH$, then $(p_1 \leq_p p_3)\in PH$. 

---

www.security-conference.org
Because the purposes are organized into a hierarchical tree, when a purpose $p$ is permitted, all the offspring purposes of $p$ according to the tree are permitted too. We define a set $P$ to represent all permitted purposes implied by $p$ and $p$.

**Definition 3.** If $p \in P$ is given, then $P = p \cup \{ p' | \forall p' \in P, (p \leq p') \in PH \}$. We call $P$ is the range of purpose $p$.

**Definition 4.** An intent purpose is denoted as $ip$, where $ip \in P$. An intent purpose of a data $dat$, $ip_{dat}$, represents the reasons and guidelines for which the data $dat$ can be released. Here $ip_{dat}$ is specified by the data owner. The range of the intent purpose is denoted as $IP$.

**Definition 5.** A designed purpose is denoted as $dp$, where $dp \in P$. A designed purpose of an object (workflow, logical task, or atomic task), $dp_o$, where $o \in \{ \text{workflow, logicalTask, atomicTask} \}$, represents the reasons and/or guidelines for which the input data will be processed by the object. Here the $dp_o$ is specified by the designer of the object. The range of the designed purpose is denoted as $DP$.

**Definition 6.** A use purpose is denoted as $up$, where $up \in P$. A use purpose, $up_i$, where $i \in \{ \text{workflow, logicalTask, atomicTask} \}$, describes the reasons and/or guidelines for which the instance $i$ actually uses the data. The range of the use purpose is denoted as $UP$, which is decided by the object instance $i$.

A designed purpose of a workflow specifies a set of general reasons and/or guidelines that are applied to all input data of the workflow, but the use purpose of the workflow is specified for a particular piece of sensitive data. In most cases, the range of the use purpose of the workflow for a particular data is a subset of the range of the designed purpose of the workflow, because the sensitive data may not be processed by all tasks involved in the workflow and only passes through partial of the workflow.

A designed purpose of a logical task specifies a set of general reasons and/or guidelines that are applied to all input data of the logical task. The use purpose of the logical task totally depends on a (sub)workflow that is chosen to implement/replace the logical task. Generally speaking, in order to protect the sensitive data, the range of the designed purpose of the logical task should be a super-set of the range of the use purpose of the workflow implementing the logical task.

A designed purpose of an atomic task also specifies a set of general reasons and/or guidelines of all input data of the atomic task. At the workflow level, an atomic task is treated as a black box, and no detail of using the data is available to the users working on the workflow level. Therefore, for an atomic task, the use purpose of the atomic task equals its designed purpose.

**b. Extended Process Grammar**

A workflow process definition is a formal representation of a business process. According to the Process Grammar (Boldwin 1995), a workflow is defined as a set of tasks, including logical tasks and atomic tasks, and data specifications. In order to support purpose constraints, we extend the
original Process Grammar definitions. The extended definitions of workflow, logical task, atomic task and data specification are defined as follows:

**Definition 7.** A **workflow** is defined as a tuple:

$$w(\text{DAT}, \text{LT}, \text{AT}, \text{IN}, \text{OUT}, dp)$$

where, \text{DAT} is a set of data specifications, \text{DAT} \subseteq \text{DAT}, \text{LT} is a set of logical tasks, \text{LT} \subseteq \text{LT}, \text{AT} is a set of atomic tasks, \text{AT} \subseteq \text{AT}, \text{IN} is a set of directed links pointing from data specifications to the workflow \( w \), \text{IN} \subseteq \text{IN}, \text{OUT} is a set of directed links pointing from the workflow \( w \) to data specifications, \text{OUT} \subseteq \text{OUT}, dp is the designed purpose of \( w \), dp\( \in \text{P} \). The global set of workflows is defined as \text{W}.

**Definition 8.** A **logical task** is defined as a tuple:

$$lt(\text{IN}, \text{OUT}, W, dp)$$

where, \text{IN} is a set of directed links pointing from data specifications to \( lt \), \text{IN} \subseteq \text{IN}, \text{OUT} is a set of directed links pointing from \( lt \) to data specifications, \text{OUT} \subseteq \text{OUT}, \text{W} is a set of workflows that can be used to implement/replace \( lt \), and \text{W} \subseteq \text{W}, dp is the designed purpose of \( lt \), and dp\( \in \text{P} \). The global set of logical tasks is defined as \text{LT}.

**Definition 9.** An **atomic task** is defined as a tuple:

$$at(\text{IN}, \text{OUT}, TL, dp)$$

where, \text{IN} is a set of directed links pointing from data specifications to \( at \), and \text{IN} \subseteq \text{IN}, \text{OUT} is a set of directed links pointing from \( at \) to data specifications, \text{OUT} \subseteq \text{OUT}, \text{TL} is a set of tools that can be used to carry out \( at \), \text{TL} \subseteq \text{TL}, dp is the designed purpose of \( at \), dp\( \in \text{P} \). The global set of atomic tasks is defined as \text{AT}.

**Definition 10.** A **data specification** is defined as a tuple:

$$\text{dat} (\text{meta} \_\text{data}, \text{content}, ip)$$

where, \text{meta} \_\text{data} defines the categories of data stored in \text{dat}, \text{content} is the real information stored in \text{dat}, \text{ip} is the intent purpose of \text{dat}. The global set of data specifications is defined as \text{DAT}.

**Definition 11.** An **input link** is defined as an ordered pair:

$$in < \text{dat}, o>$$

where \text{dat}, o\( \in \text{DAT} \) is a data specification that is an input of object \( o \), o\( \in \{ \text{W}, \text{LT}, \text{AT} \} \). The global set of input links is defined as \text{IN}.

**Definition 12.** An **output link** is defined as a ordered pair:

$$out < o, \text{dat}>$$

where \text{dat}, o\( \in \text{DAT} \), is a data specification that is an output of object \( o \), o\( \in \{ \text{W}, \text{LT}, \text{AT} \} \). The global set of output links is defined as \text{OUT}.
Definition 13. A tool, $tl$, refers to an application (software or hardware), which is used to actually process the data. $tl \in TL$, where $TL$ is the global set of tools.

c. Workflow-level Privacy Protection

In this section I will formally describe the algorithm for discovering the use purpose of a workflow instance.

In an outsourcing environment, in order to protect sensitive data the main aspect, which needs to be examined before releasing the data to an instance of workflow, is the use-purpose. The data must be used in the right way as it is expected, in other words, the range of intent purpose has to be preserved as the data is (re)distributed among the companies in the outsourcing environment.

The decision to release data to an object (workflow, logical task or atomic task) instance is:

$$Release(dat, o) = PreservePurpose(dat, o)$$

Where, data $dat$, $dat \in DAT$, is sensitive, and object $o$ is an instance, $o \in \{W, LT, AT\}$.

When data is released from a keeper to a receiver, the keeper specifies the purpose $p$ of the data and sends the purpose to the receiver together with the data. Actually there are two ways to generate the purpose $p$, one way is the keeper specifies the purpose arbitrarily; the other way is the receiver proposes a purpose, and then the keeper approves it. Our system focuses on the second way. In our system, the receiver proposes the use purpose $up$ and the keeper approves and sends it back with data. In order to protect data privacy along the outsourcing chain, we have the following rule:

**Rule 2.** Suppose data $dat$, $dat \in DAT$, is sensitive, and $dat$ is an input of object $o$, $PreservePurpose(dat, o) = TRUE$, if $P_{up} \subseteq P_{ip_{dat}}$ is true, where $P_{ip_{dat}}$ is the range of $ip_{dat}$ and $ip_{dat}$ is the intent purpose of data $dat$, $P_{up}$ is the range of $up_o$, and $up_o$ is the use purpose of the instance of $o$, $o \in \{W, LT, AT\}$, for the sensitive data $dat$.

d. Algorithm for Use-Purpose Discovering

i. The Generation of the Use Purpose Hierarchy

The following steps are used to generate the use purpose $up_w$ of a workflow instance $w(DAT_w, LT_w, AT_w, IN_{w}, OUT_{w}, dp_{w})$ for the data $dat$ with sensitive information meta_data category $c$.

1. Let $DAT_{sen}$ be a set of data specifications which contain sensitive information.

   $$DAT_{sen} = \{dat | dat \in DAT_w, c \in meta\_data\}$$

2. Let $LT_{sen}$ be a set of logical tasks which take sensitive data as input.

   $$LT_{sen} = \{lt | \exists in<dat, lt> \in IN_{lt}, dat \in DAT_{sen}, lt \in LT_w\}$$

3. Let $AT_{sen}$ be a set of atomic tasks which take sensitive data as input.

   $$AT_{sen} = \{at | \exists in<dat, at> \in IN_{at}, dat \in DAT_{sen}, at \in AT_w\}$$
(4) Let \( \text{UP}_w \) and \( \text{UPH}_w \) be the use purpose set and hierarchy of the instance \( w \). The following steps are used to generate the set and the hierarchy.

a) Let \( up_w \) be the use purpose of \( w \). The initial set of use purposes is \( \text{UP}_w = \{up_w\} \).

b) Let \( dp_w \) be the designed purpose of \( w \). The initial use purpose hierarchy is: 
\[ \text{UPH}_w = \{(up_w \leq dp_w)\} \]

c) Let \( up_{at} \), when \( at \in \text{AT}_{\text{sem}} \), be the use purpose of an atomic task that takes the sensitive data as an input. Then,
\[
\text{UP}_w = \text{UP}_w \cup \{up_{at}\}
\]
\[
\text{UPH}_w = \text{UPH}_w \cup \{(dp_w \leq up_{at})\}
\]

d) Generating the use purpose of the atomic task \( at \). Let \( dp_{at} \) be the designed purpose of an atomic task instance \( at \), then
\[
\text{UP}_w = \text{UP}_w \cup \{dp_{at}\}
\]
\[
\text{UPH}_w = \text{UPH}_w \cup \{(up_{at} \leq dp_{at})\}
\]

e) Let \( up_{lt} \), when \( lt \in \text{LT}_{\text{sem}} \), be the use purpose of a logical task that takes the sensitive data as an input, then,
\[
\text{UP}_w = \text{UP}_w \cup \{up_{lt}\}
\]
\[
\text{UPH}_w = \text{UPH}_w \cup \{(dp_w \leq up_{lt})\}
\]

f) Generating the use purpose of a logical task \( lt \). Let \( dp_{lt} \) be the designed purpose of the logical task \( lt \). Let \( w' \) is a workflow instance used to implement the logical task, and \( up_{w'} \) is the use purpose of the workflow instance \( w' \).

i. If the workflow instance \( w' \) is unknown, then
\[
\text{UP}_w = \text{UP}_w \cup \{dp_{lt}\}
\]
\[
\text{UPH}_w = \text{UPH}_w \cup \{(up_{lt} \leq dp_{lt})\}
\]

ii. If the workflow instance \( w' \) is known, then
\[
\text{UP}_w = \text{UP}_w \cup \{up_{w'}\}
\]
\[
\text{UPH}_w = \text{UPH}_w \cup \{(up_{lt} \leq up_{w'})\}
\]

(5) For \( up_{w'} \), repeat step 1 to 4 until no further purposes can be inserted.

(6) Finally \( \text{UP}_w \) and \( \text{UPH}_w \) are generated.

ii. Trim the Purpose Hierarchy
Figure 7. Original Use Purpose Hierarchy

Figure 7 shows the original use purpose hierarchy of workflow instance $w$ for the sensitive data $dat$. In order to compare the use purpose of $w$ with the intent purpose $ip_{dat}$ of $dat$, first, all use purposes in this hierarchy need to be resolved, because they are unknown; second, this hierarchy needs to be simplified. The algorithm used to trim the hierarchy is shown below: (Note: this algorithm works bottom up.)

(1) For an atomic task $at$, $up_{at}$ is simply replaced by $dp_{at}$.

$$
UPH_w = (UPH_w - \{dp_w \leq_P up_{at}\}) \cup \{(dp_w \leq_P dp_{at})\}
$$

$$
UP_w = UP_w - \{up_{at}\}
$$

(2) For a logical task $lt$, there are three ways to compute $up_{lt}$. Suppose the workflow instance $w'$ used to implement $lt$.

a) If the workflow $w'$ is unknown, $up_{lt}$ is replaced by $dp_{lt}$.

$$
UPH_w = (UPH_w - \{dp_w \leq_P up_{lt}\}) \cup \{(dp_w \leq_P dp_{lt})\}
$$

$$
UP_w = UP_w - \{up_{lt}\}
$$

b) If the workflow $w'$ is known, $up_{lt}$ is replaced by $up_{w'}$.

$$
UPH_w = (UPH_w - \{dp_w \leq_P up_{lt}\}) \cup \{(dp_w \leq_P up_{w'})\}
$$

$$
UP_w = UP_w - \{up_{lt}\}
$$

c) If the sensitive data is not used by the workflow $w'$, then $up_{lt}$ is removed.

$$
UPH_w = UPH_w - \{(dp_w \leq_P up_{lt}), (up_{lt} \leq_X up_{w'})\}
$$

$$
UP_w = UP_w - \{up_{lt}, up_{w'}\}
$$
Figure 8. Trimmed Use Purpose Hierarchy

Figure 8 shows the trimmed use purpose hierarchy for workflow \( w \) for the sensitive data \( dat \). The next step is to simplify this hierarchy. Here we assume that the use purpose of the workflow instance \( w' \) implementing a logical task is known. The following are rules to simplify the hierarchy.

**Rule 3.** Suppose \( PH \) is the globally accepted purpose hierarchy, \( UPH_w \) and \( UP_w \) are the use purpose hierarchy and set of workflow instance \( w \) for a sensitive data \( dat \), \( up_w \) is the use purpose and \( dp_w \) is the designed purpose. \( UP_w \) is the range of \( up_w \) and \( UP_{dpw} \) is the range of \( dp_w \).

If \( (dp_w \leq p_i) \in PH \), where \( \forall p_i \in UP_w - \{up_w, dp_w\} \) and \( (dp_w \leq p_i) \in UPH_w \), then

\[
UP_w = UP_{dpw}
\]

else

\[
UP_w = UP_{dpw} \cup \{p_i \mid \forall p_i \in UP_w - \{up_w, dp_w\}, (dp_w \leq p_i) \notin PH\}
\]

Based on the globally accepted purpose hierarchy, if the range of the use purpose of every sub-step (logical task, atomic task and sub-workflow) is smaller or equal to the designed purpose of the top-level workflow \( w \), the use purpose of the workflow instance will be the designed purpose of the top-level workflow. If the ranges of the use purposes of the sub-steps are beyond the designed purpose of the top-level workflow, the use purpose of the workflow instance is the union of the designed purpose of the top-level workflow and other purposes beyond the designed purpose.

e. Making the Decision to Release Data

After generating the use purpose of the workflow instance \( w \), the sender can make the decision to release the data or not to the receiver which executes the workflow instance \( w \) to process the data.

For data \( dat \) with an intent purpose \( ip_{dat} \), we have

\[
P_{ip} = ip_{dat} \cup \{p \mid \forall p \in P, (ip_{dat} \leq p) \in PH\}
\]

If \( UP_w \subseteq P_{ip} \) is true, then

\[
Release(dat, w) = \text{PreservePurpose}(dat, w) = \text{TURE}
\]

else
Release(dat, w)=PreservePurpose(dat, w)=FALSE

If the range of use purpose $UP_w$ of the workflow is a subset of the intent purpose $P_ip$ of the data, it means that the data is used in a right way, and then the keeper will send the data to the receiver. The data will be processed through the workflow $w$ by the receiver. At the same time $UP_w$ becomes the new $P'_ip$ for data on the receiver side. Once the receiver needs to send the data to others, $P'_ip$ will be checked before sending the data. In such a way, along the outsourcing chain a new intent purpose $P'_ip$ is always a subset of the original $P_ip$, which means the range of the original intent purpose from the owner will be preserved through the chain, so that the data privacy is also protected through the chain.

6. Conclusion

Outsourcing helps enterprises cut operating costs, but it also puts the enterprise's sensitive data in danger. In this paper we outlined a framework for managing the privacy of data through the outsourcing chain and proposed a formal model and a mathematic algorithm to compute the use purpose. We use existing workflow management systems combined with the concept of purpose to select the “right” workflow at runtime to ensure the protection of privacy.

The uniqueness of our framework is to control the (re)distribution of data among activities to achieve privacy protection. By integrating the support of purpose into workflow management systems, the safety of the distribution and redistribution of private data will be ensured.

The use-purpose discovering algorithm was implemented. The application takes the workflows, which are defined in terms of the extended Process Grammar, and automatically generates the use purpose of a workflow instance. The future work on this topic is to establish a set of standards of purpose hierarchies that can be used by the workflow designers to describe purposes for the tasks and workflows. How to reason purposes from workflows directly is another interesting topic.
References


