World Wide Web (2011) 14:407–430
DOI 10.1007/s11280-011-0114-8

Privacy-aware access control with trust management in web service

Min Li · Xiaoxun Sun · Hua Wang · Yanchun Zhang · Ji Zhang

Received: 29 June 2010 / Revised: 27 December 2010 /
Accepted: 24 January 2011 / Published online: 3 February 2011

Abstract With the significant development of mobile commerce, privacy becomes a major concern for both customers and enterprises. Although data generalization can provide significant protection of an individual's privacy, over-generalized data may render data of little value or useless. In this paper, we devise 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, we propose a privacy-aware access control model in web service environments. We also analyze how to manage a valid access process through a trust-based decision and ongoing access control policies. The extensive experiments on both real-world and synthetic data sets show that the proposed privacy aware access control model is practical and effective.

Keywords access control · privacy protection · generalization boundary

M. Li (✉) · H. Wang · J. Zhang
Department of Mathematics & Computing, University of Southern Queensland, Toowoomba, QLD, Australia
e-mail: limin228@gmail.com

H. Wang
e-mail: hua.wang@usq.edu.au

J. Zhang
e-mail: ji.zhang@usq.edu.au

X. Sun (✉)
Australian Council for Educational Research, Camberwell, VIC, Australia
e-mail: sun@acer.edu.au

Y. Zhang
School of Engineering and Science, Victoria University, Melbourne, VIC, Australia
e-mail: Yanchun.Zhang@vu.edu.au

1 Introduction

Advances in wireless technology have stimulated rapid developments in electronic commerce (e-commerce) via the use of mobile devices. E-commerce transactions conducted through radio-based wireless devices are called mobile commerce (also known as m-commerce or mobile e-commerce). Mobile commerce can extend current Internet sales channels into more immediate and personalized mobile environment. While current information technology enables people to carry out their business virtually at any time in any place, it also provides the capability to store various types of information that users reveal during their activities. The use of innovative knowledge extraction techniques combined with advanced data integration and correlation techniques make it possible to automatically extract a large body of information from available databases and from a large variety of information repositories available on the web [8, 16]. Privacy issues are exacerbated by the Internet which makes it easy for new data to be automatically collected and added to databases [15, 23, 24].

Changes in the legislation around the world and growing consumer attention have changed attitudes towards security and privacy concerns for database systems. This coincides with a substantial body of research on approaches for managing the negotiation of personal information among customers and enterprises [2, 17, 21]. At the heart of protecting the privacy is the principle of transparency. Transparency means that when enterprises store data about customers they should disclose to customers what data is being collected and how it is to be used; i.e. for what purpose data is being used and how it is maintained. Starting from the landmark proposal for Hippocratic databases [3], most privacy-aware technologies use purpose as a central concept around which privacy protection is built. Byun and Bertino [5] proposed a model based on a typical life-cycle of data concerning individuals. The use of data generalization[1] helps to significantly increase the comfort level of the data providers. For example, many individuals may not be comfortable with their date of birth being used. Suppose the enterprise promises its customers that this information will be used only in a generalized form; e.g. (08/20/1980) will be generalized to a less specific value (08/1980). This assurance can provide much comfort to many customers and the ability to limit the level of allowed generalization could be valuable in terms of privacy. However over-generalization of data could make it useless; for instance, when address information, such as 14 Regent Street, Toowoomba, Queensland, Australia, is used for some specific data analysis tasks in relation to States in Australia, then the state "Queensland" should be the maximal allowed generalization value. Therefore, the address information generalized beyond the state could be useless. Hence the issue is how to determine whether or not a certain generalization strategy provides a sufficient level of privacy and usability.

[1] Data generalization refers to techniques that "replace a value with a less specific but semantically consistent value."

One of the most important challenges is that the comfort level of privacy varies from individual to individual, and this requires incorporating generalization techniques with sufficient levels of privacy and usability.

Privacy concerns are fueled by an ever increasing list of privacy violations, ranging from privacy accidents to illegal actions. Many people are aware that giving personally identifiable information (PII) to organizations may result in the data being used in ways the person never intended. Individuals are becoming more reluctant to carry out business and transactions online potentially leading to many enterprises losing a considerable amount of their profits. Thus, another daunting challenge to ensure wide diffusion of mobile commerce concerns trust in mobile commerce. Lack of consumer trust is the most significant long-term barrier for e-commerce, as well as for mobile commerce. Although mobile devices are more convenient for “anytime shopping”, it has some unique features and characteristics that hinder the development of consumer trust. To become a viable means of doing business, mobile commerce must overcome the problem of user distrust.

We emphasize that the above issues cannot be easily achieved by traditional access control models. Traditional access models, such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), and Role Based Access Control (RBAC) [9, 13, 15], are fundamentally inadequate. The first reason is that privacy policies are concerned with which data object is used for which purpose(s), while traditional access control models focus on which user is performing which action on which data object. Another difficulty is how to make the access control technology in a trustworthy fashion, when the data provider and the requester are unknown to each other. We believe that the availability of new generation access control mechanisms is an important requirement.

In this paper, we devise a generalization boundary technique to balance privacy and information utilization, satisfying the requirements of both data providers and data users. Moreover, we propose a privacy-aware access control model, where formalized authorizations are defined relating the permissible usage and specific generalization levels. The trust-based decision policy and ongoing access control policy combined together create a secure protection system. Further, our model provides a much finer level of control as the access control decision is based on the question of “how much information can be allowed for a certain user”, rather than “is information allowed for a certain user or not”. Trust-based decision and ongoing access control mechanisms are designed to manage a valid access process at the pre-access and ongoing-access stages, respectively. Finally, we describe the state transition architecture of the privacy-aware access control model to demonstrate how the model works in practice. Proof-of- concept experimental studies confirm that our proposed privacy-aware access control model with generalization boundaries is practical and effective

The remainder of the paper is structured as follows. In Section 2, we describe the motivation of the paper. We specify the generalization boundaries and propose the privacy-aware access control model in Section 3. In Sections 4 and 5, we discuss how to manage a valid access process through trust-based decision and ongoing access control mechanisms. In Section 6, we evaluate the proposed privacy-aware access control model over two data sets. We provide a brief survey of related work in Section 7 and conclude the paper in Section 8.

Table 1 Privacy information and metadata.

Name		Address		Income		Admin	Marketing	Delivery
L	Alice Park	L	123 First St., Seattle, WA	L	45,000	{L, M, H}	{M, H, H}	{M, M, M}
M	A. Park	M	Seattle, WA	M	40K–60K			
H	A.P.	H	WA	H	Under 100K			

2 Motivation

Following [5], the actual data items[2] are preprocessed before being stored. The pre-processing takes the following form: Each data item is generalized and stored according to a multilevel organization, where each level corresponds to a specific privacy level. Intuitively, data for a higher privacy level requires a higher degree of generalization. Let us briefly describe the terminologies used in this process:

- *Data Provider*: Data provider refers to the subject to whom the stored data is related. We denote S as the set of data providers.
- *Data Users*: Data users are individuals who access or receive data. Data users are required in a privacy context, as privacy policies will depend on the relationship between the individual requesting data and the individual to whom the data is related to. For example, one type of data users might be *physician* while another might be *primary care physician*. We denote U as the set of data users.
- *Privilege*: Some privacy policies make distinctions about who can perform activities based on the action being performed. For example, a policy might state that anyone in the company can *create* a customer record, but that only certain data users are allowed to *read* that record. We denote $Priv$ as the set of privileges.
- *Purpose*: Data access requests are made for a specific purpose or purposes. This represents how the data is going to be used by the recipient. For example, the data may be used for *Marketing* or *Delivery* purposes. We denote P as the set of purposes.
- *Generalization Level*: Generalization level refers to what extent the data items have been generalized. We denote GL as the set of private levels, which consists of Low, Medium, High, and Maximal generalization level, denoted as L, M, H and ML. For example, a *Low* generalization level on *Address* means that the address information can be used without any modification.

Table 1 illustrates some fractional records and privacy requirements stored in a conceptual database relation. Note that each data item is stored at three different privacy levels, *Low*, *Medium*, *High*. Take the *address* data as an example: the entire address is regarded as *Low*, city and state are at *Medium* and state at *High*. Admin and Marketing are metadata columns storing the set of privacy levels of data for *Admin* and *Marketing* purposes respectively. Further, a data provider submits his/her privacy requirements, which specify permissible usages of each data item and a level of privacy for each usage. For instance, {M, H, H} under Marketing indicates that for the *Marketing* purpose data users can only access *Name* at the *Medium* privacy level while accessing *Address* and *Income* at the *High* level.

[2] Data item refers to the type of data being collected (i.e., attributes), such as *Name*, *Address*. In this paper, we denote D as the set of data items.

Table 2 Private information for *Delivery* purpose.

Name	Address	Income	Delivery
A. Park	Seattle, WA	40K–60K	{M, M, M}

We can see that the access to each data item is strictly governed by the data provider's requirements. Before data access, authorizations on each data item have already been granted through the permissible usage requirements. However, different people may have different feelings about their information being used for some purposes. For instance, some consumers may feel that it is acceptable to disclose their purchase history or browsing habits in return for better services; others may feel that revealing such information violates their privacy. Differences in individuals suggest that access control models should be able to maximize information utility, which may be neglected by data providers although wanted by data users. For example, if a data provider selects {M, M, M} on *Name*, *Address*, *Income* for *Delivery* purpose, (i.e., the data user has been authorized to access *Name*, *Address*, and *Income* only in medium level shown in Table 2) then, the information could be useless for the data user who wants to fulfill the delivery purpose because full name and address are necessary information for delivery. Further, the {M, M, M} selection may increase the chance of disclosure of the unnecessary information *Income* since the more people who know, the more likely it would be disclosed. Authorizations incurred by this selection could not protect data privacy (e.g., *Income*, to some degree) nor maintain data usability.

To solve this problem, we need metrics that methodologically measure the privacy and usability of generalized data. It is necessary to devise efficient generalization techniques that satisfy the requirements of both data providers and data users. In this paper, we propose a privacy-aware access control model with generalization boundaries, which can maximize data usability and minimize privacy disclosure. In particular, we

- Formalize the authorizations with the specific purpose and generalization levels specified on each data item.
- Propose a trust-based decision policy with trust evaluation techniques to handle access security with regard to a requester's trust before data access.
- Design authorization and access functions to handle access security with regard to the retention period and generalization level during data accessing.
- Study the state transition of our proposed privacy-aware access control model and illustrate how the model works in practice.
- Evaluate our proposed access control model on both real-life and synthetic data sets to show its efficiency and effectiveness.

3 Privacy-aware access control model

By using data generalization, data providers can specify their privacy requirements using a generalization level for each data item. Data for a higher privacy level requires a higher degree of generalization; i.e., each privacy level is accompanied with a generalization level. However, over-generalized data may render data of little value or useless. In this section, we introduce a privacy-aware access control model with generalization boundaries.

3.1 Generalization boundary

In order to specify a generalization boundary, we introduce the concept of a maximum allowed generalization level that is associated with each data item. This concept is used to express to what extent the data user thinks the data item could be generalized, such that the resultant generalized data item would still be useful. Limiting the level of generalization for the data item is necessary for various usage of the data. For instance, when data related to Australian states is used for some specific analysis tasks, the data user will select the level corresponding to the states as the maximal allowed generalization level. Address information generalized beyond the Australia state level could be useless. In this case, the only solution would be to ask the data provider to make a decreased level of generalization until the generalized data satisfies the maximum allowed generalization level requirement (i.e., no address is generalized further than the Australian state).

Definition 1 Let D be the set of data attributes and P be the set of purposes. For each data attribute $d \in D$ and purpose $p \in P$, the **maximum allowed generalization level** of d under purpose p, denoted by $MAGLel(d, p)$, satisfies that the data attribute d is permitted to be generalized only up to $MAGLel(d, p)$.

We assume that the generalization level is equal to the privacy level in this paper. The maximal generalization level, denoted ML, corresponds to generalizing a data value to $*$. For simplicity of discussion, we only consider the generalization levels: low (L), medium (M), high (H) and (ML). For example, if $D = \{$Name, Address, Income$\}$, $P = \{$Admin, Marking, Delivery$\}$, then we can define the maximum allowed generalization level of *Name* under purpose *Delivery*, $MAGLel(Name, Delivery) = L$.

Note that the maximum allowed generalization level of the data could be different for different purposes. For example, the maximum allowed generalization level of *Address* could be *Low* for *Delivery* purpose, whereas it may be *High* for *Marketing* purpose. Usually, for a certain purpose, the data user only has generalization restrictions for some necessary data items; e.g., there should be restrictions on *Name* and *Address* for *Delivery* purpose but no restrictions on *Income*. If for a particular data item there are no any restrictions with respect to its generalization, then the maximal generalization level ML is specified for the usage of this data. In this case, the requirement of providing sufficient privacy and usability is satisfied by the following description.

Definition 2 Let P be the set of access purposes and D be the set of data items, for each purpose $p \in P$, the set $N_p \subseteq D$ denotes all necessary data attributes to fulfill the purpose p. The **privacy-aware generalization boundaries** for p satisfies the following:

- for $\forall d \in N_p$, the data attribute d is permitted to be generalized only up to $MAGLel(d, p)$;
- for $\forall d \notin N_p$ and $d \in D$, the data attribute d is permitted to be generalized up to ML.

Table 3 Generalization boundaries for *Delivery* purpose.

Name		Address		Income		Delivery
L	Alice Park	L	123 First St., Seattle, WA	L	45,000	{MAGLel(Name, Delivery), MAGLel(Address, Delivery), ML}
M	A. Park	M	Seattle, WA	M	40K–60K	
H	A.P.	H	WA	H	Under 100K	
ML	*	ML	*	ML	*	

For instance, if D = {Name, Address, Income} and P = {Admin, Marking, Delivery}, then since the full name and address are necessary to fulfill the *Delivery* purpose, N_{Delivery} = {Name, Address}. Table 3 shows the example of privacy-aware generalization boundaries for the *Delivery* purpose. Because of $Name, Address \in N_{\text{Delivery}}$, the generalizations on *Name* and *Address* are only permitted up to $MAGLel(Name, Delivery)$ and $MAGLel(Address, Delivery)$ (i.e., *Low* and *Low*), respectively. On the other hand, for *Income*, there are no requirements with respect to its generalization, since $Income \notin N_{\text{Delivery}}$, so the maximal generalization level ML is specified for the usage of *Income*. The information obtained by the data user is shown in Table 4.

The above example shows that our proposed generalization boundary strategy can maximize data usability while, at the same time, minimizing disclosure of data privacy. Moreover, the specific generalization boundaries actually describe the permissible usage of each data item, and the permissible usage further grants the data user to access each data item from a specific generalization level. Such a finer level access control could satisfy the requirements of both data providers and data users. Now the issue is how to build a formal access control model with specific generalization boundaries that can balance data privacy and usability. We discuss this question in detail in the next section.

3.2 Privacy-aware authorizations

Authorization is the act of checking to see if a data user has the proper permission to access the particular data or perform a particular action. In addition to the traditional authorization factors, data items, data users and privileges, all authorizations in this paper are extended to include the specific purpose and generalization level on each data item.

Moreover, personal information is retained only as long as necessary for the fulfillment of the purpose for which it has been collected. Retention period refers to how long the information is stored. For example, if the retention period for *Name* is one month, the name information can only be retained for one month. We use time intervals to describe retention period, e.g., [12/02/2008, 12/03/2008]. We denote T as the set of time intervals. If a certain data item was collected for a set of purposes, it is kept for the limited retention period of the purpose. We refer to an authorization together with its usage time as a generalized authorization. A time

Table 4 Ideal information for *Delivery* purpose.

Name	Address	Income	Delivery
Alice Park	123 First St., Seattle, WA	*	{L,L,ML}

interval is also associated with each authorization, imposing lower and upper bounds to the potential usage.

Definition 3 A generalized authorization is a 6-tuple $(t, u, d, priv, p, gl)$, where $t \in T, u \in U, d \in D, priv \in Priv, p \in P, gl \in GL$.

A tuple $([t_a, t_b], u, d, priv, p, gl)$ states that the data user u has been authorized to perform $priv$ on the data item d in the generalization level gl for the purpose p in the time interval $[t_a, t_b]$. We denote AU as the set of temporal generalized authorizations and $\sigma_{au}(*)$ as the function used to extract the element(s) $*$ in an authorization $au \in AU$. A temporal generalized authorization $au = ([12/06/2008, 10/08/2008], Tom, income, read, admin, M)$, means that between June 12, 2008 and Augest 10, 2008, *Tom* was authorized the privilege to *read* the customer's *income* at the generalization level *Medium* for the *admin* purpose.

3.3 Authorization specification

An authorization is an approval of a particular mode of access to one or more objects in the system. Observe that in a group of authorization assignments, two authorization assignments may interact with each other when they share the same user, same data and same action. Purposes mentioned in an authorization naturally have a hierarchical relationship among them. For instance, a group of purposes such as direct-marketing and third-party marketing can be represented by a more general purpose, marketing. More specific authorizations may deal with more specific purposes that fall under the domain of a high-level purpose. This suggests that purpose can be organized according to the hierarchical relations to simplify their management. Mathematically, a purpose hierarchy is represented as a tree. Each purpose (except the root purpose) has exactly one parent purpose and there are no cycles. A parent node represents a more general purpose than those represented by its children nodes. Thus the hierarchy of purposes can be intended as a grouping of more particular purposes into more general ones. The same argument also could apply to generalization levels. Generalization refers of replacing the actual value of the attribute with a less specific, more general value which is faithful to the original [18–20]. For example, the name 'Carol Jones' can be generalized to a less specific value 'C. Jones' or further generalized to 'C.J.'. As for purposes hierarchies, a generalization hierarchy is represented as a tree structure. The meaning associated with the generalization hierarchy is analogous to the one mentioned for purpose hierarchies. Here, we use operation $\geq$ to indicate the dominance relationship in the purpose hierarchy and generalization hierarchy.

Explicit (implicit) authorization The introduction of hierarchies of purpose and generalization level with a retention period lead us to get two types of authorizations, called explicit authorizations and implicit authorizations.

Definition 4 Let $au_1 = (t_1, u_1, d_1, priv_1, p_1, gl_1)$ and $au_2 = (t_2, u_2, d_2, priv_2, p_2, gl_2)$ be two authorization in AU. We say that au_1 is an explicit authorization of au_2 (or au_2 is an implicit authorization of au_1) only if one of the following conditions satisfies:

- $(t_1 \supseteq t_2) \wedge (u_1 = u_2) \wedge (d_1 = d_2) \wedge (priv_1 = priv_2) \wedge (p_1 = p_2) \wedge (gl_1 = gl_2)$

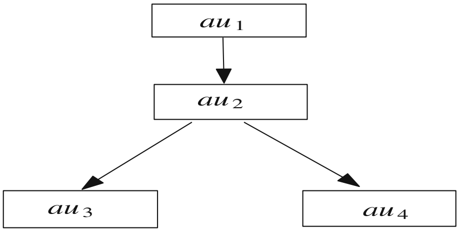

Figure 1 Authorization tree.

- $(t_1 = t_2) \wedge (u_1 = u_2) \wedge (d_1 = d_2) \wedge (priv_1 = priv_2) \wedge (p_1 \geq p_2) \wedge (gl_1 = gl_2)$
- $(t_1 = t_2) \wedge (u_1 = u_2) \wedge (d_1 = d_2) \wedge (priv_1 = priv_2) \wedge (p_1 = p_2) \wedge (gl_1 \geq gl_2)$

For example, let $au_1, au_2, \ldots, au_9$ be authorizations, where

$$au_1 = ([9AM, 5PM], Tom, email, read, Marking, M),$$
$$au_2 = ([9AM, 3PM], Tom, email, read, Marking, M),$$
$$au_3 = ([9AM, 3PM], Tom, email, read, Third - party\ Marketing, M)$$
$$au_4 = ([9AM, 3PM], Tom, email, read, Third - party\ Marketing, H),$$

then they can be represented as a tree (Fig. 1).

Conflicting authorizations Complex environments, such as large enterprises, usually have to comply with complex security and privacy policies. As such, it is possible that the more complex a security policy is, the larger the probability that such policy contains inconsistent and conflicting parts is. In particular, authorization assignments could conflict because of new requirements, new regulations, or just human mistakes.

Consider the following authorization assignments:

au_1 =([9AM, 5PM], Bank manager, loan, approve, Marketing, Low)

au_2 =([9AM, 5PM], Bank manager, loan, fund, Marketing, Low).

Notice that there are different privileges related to the same user working on the same data in the generalization level for the purpose in the time interval. A tricky issue here is that the privileges of approving a loan in a bank and that of funding a loan are conflicting. Therefore, these two authorizations conflict with each other since they have conflicting privileges.

Definition 5 Let $au_1 = (t_1, u_1, d_1, priv_1, p_1, gl_1)$ and $au_2 = (t_2, u_2, d_2, priv_2, p_2, gl_2)$ be two authorization in AU. We say that au_1 and au_2 are conflicting only if $priv_1$ and $priv_2$ are conflicting.

4 Access control process

After each data is granted with authorizations according to different purposes, an access request is needed to access the data items. In this paper, we assume that each access request is associated with an access time and a specific purpose. It is not trivial for a system to correctly infer the purpose of a query as the system must correctly deduce the actual intention of database users.

Definition 6 An access request is a 5-tuple $(t, u, d, priv, p)$ where $t \in T$ is the time when the access is requested, $u \in U$ is the data user who requires the access, $d \in D$ is the data item to be accessed, $priv \in Priv$ is a privilege exercised on the data, and $p \in P$ is the purpose for which the data is going to be used.

The tuple $([t_a, t_b], u, d, priv, p)$ states that the data user u requests to perform $priv$ on the data item d for purpose p in the time interval $[t_a, t_b]$. We denote R as the set of access requests and for an access request $r \in R$, $r(*)$ refers to the element(s) $*$ in an access request r. For example, the access purpose $r =$ ([10/07/2008, 20/07/2008], *Tom*, *income*, *read*, *admin*) means that between July 10, 2008 and July 20, 2008, *Tom* requests to *read* the customer's *income* information for the *admin* purpose. Here, $r(t)$ refers to the time interval [10/07/2008, 20/07/2008].

Under a request, traditional access process refers to a general way of controlling access to data items and makes authorization decisions based on the identity of the resource requester. Unfortunately, when the resource owner and the requester are unknown to one another, access control based on identity may be ineffective. Access control technology can be used as a starting point for managing personal identifiable information (PII) in a trustworthy fashion. It is important that data items are accessed by persons who are trusted, and this requires that trust-based decisions should be made by data providers according to the data user's trust value. Next, we discuss the management of a valid access process through the trust-based decision policy.

4.1 Trust-based decision mechanism

Trust means the liability and trustworthiness of a trusted agent's behavior. There are two approaches to obtaining an agent's trust: experience by interacting with the agent, and recommendation of other agents [22]. In this paper, we evaluate the trust value in three steps (as show in Figure 2): (1) Calculate the trust value based on histories; (2) Calculate the trust value from recommendators; (3) Combine the observed trust values from histories and recommendations.

Step 1 Calculate trust based on histories.

Let m denotes the total number of transactions performed by a data user u during the given period, and $S(u, i)$ denote the satisfaction degree of the participating agent in u's i-th transaction, $S(u, i) \in [0, 1]$. If the transaction context factor of u's i-th transaction is $TF(u, i)$, then u's trust can be evaluated by direct experience as follows:

$$T_1(u) = \frac{\sum_{i=1}^{m} S(u, i) \times TF(u, i)}{\sum_{i=1}^{m} TF(u, i)} \tag{1}$$

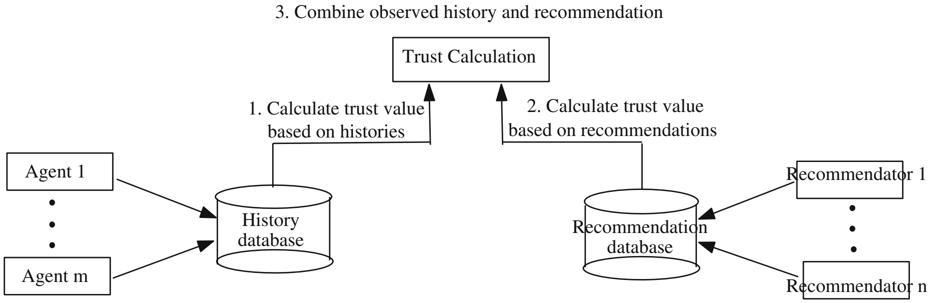
3. Combine observed history and recommendation
Trust Calculation
1. Calculate trust value based on histories
2. Calculate trust value based on recommendations
Agent 1
Agent m
History database
Recommendation database
Recommendator 1
Recommendator n

Figure 2 Trust evaluation.

Here, $TF(u, i) \in (0, 1)$ is the weight to indicate the influence of a transaction on trust value. If the value of $TF(u, i)$ is large, the transaction has more influence on trust value. Further, if a data user u behaves in a satisfactory manner in all related transactions, i.e. $S(u, i) = 1$ for every i, then u can be regarded as completely trustworthy, i.e. $T(u) = 1$.

Step 2 Calculate trust based on recommendations.

Now we consider the situation of obtaining u's trust from others' recommendation. Let n denote the total number of the recommendations, and $P(u, j) \in (0, 1)$ denote the normalized amount of satisfaction of recommendation for data user u in its j-th transaction. $TP(u, j) \in (0, 1)$ denotes the weight of j-th transaction, the recommendation-based trust value can be calculated as follows:

$$T_2(u) = \frac{\sum_{j=1}^{n} P(u, j) \times TP(u, j)}{\sum_{j=1}^{n} TP(u, j)} \tag{2}$$

Step 3 Merge history-based trust with recommendations.

Now we consider both the trust value from contacting with data user u and the trust value from others' recommendations. Choose a power $\alpha \in (0, 1)$, then we can calculate u's trust as follows:

$$T(u) = \alpha \times T_1(u) + (1 - \alpha) \times T_2(u) \tag{3}$$

The above method for calculating a data user's trust combines the trust information based on the past experiences in interacting with this data user and other's recommendations, and considers the influence of a transaction context. With this approach, we can obtain the data user's trust value, which is assigned in the range [0, 1].

Table 5 details an example on how to calculate a data user's trust value, where five transaction behaviors are recorded and recommended. The satisfaction degree from participating agents and commentators are given under $S(u, i)$. According to formulas (1) and (2), we can get the trust value $T_1(u) \approx 0.7$ based on histories and $T_2(u) \approx 0.6$ based on recommendations. Combining the two values gives the total trust value $T(u) \approx 0.68$ when the power α is chosen on 0.6.

Table 5 Example of trust calculation for data user u.

Time	$S(u, i)$	$TF(u, i)$	$P(u, i)$	$TP(u, i)$
1st	0.8	0.4	0.7	0.3
2nd	0.7	0.6	0.8	0.5
3rd	0.9	0.6	0.8	0.4
4th	0.6	0.8	0.5	0.6
5th	0.9	0.7	0.7	0.8

$$T_1(u) = \frac{\sum_{i=1}^{m} S(u,i) \times TF(u,i)}{\sum_{i=1}^{m} TF(u,i)} \approx 0.7$$

$$T_2(u) = \frac{\sum_{j=1}^{n} P(u,j) \times TP(u,j)}{\sum_{j=1}^{n} TP(u,j)} \approx 0.6$$

$$T(u) = \alpha \times T_1(u) + (1 - \alpha) \times T_2(u) \approx 0.68\ (\alpha = 0.6)$$

The data user's trust status is dynamic. An agent who once behaved well might subsequently behave maliciously. So a data user's trust value is only valid for a period of time, and it should be updated timely. Now assume that a data user requests to read a data item. The data accessible to the request normally depends on whether the requester's trust value is higher than the data provider's trust threshold for reading the data. Different accesses or services require participating users with different trust status. For example, a payment service may require that the parties are highly reliable, while ordinary file share service has a lower requirement for an agent's trust. Write access to a file needs a higher trust degree than read access to the same file, and the access to a confidential file requires a higher degree of trust than access to an ordinary file.

In our model, the data provider's trust threshold is defined as the minimum trust value for obtaining operation permission. Access is permitted only when the requester's trust degree is higher than the data provider's trust threshold. Conversely, when a data user's trust degree is less than the data provider's trust threshold for an operation, the data user will be prohibited from performing the operation.

The trust-based decision is described as follow:

Let S, U, D, $Priv$ be the set of data providers, data requester (users), data items, and operations. Then

$P_D \subseteq Priv \times D$ denotes the operations on data items
$TT\_S : S \times P_D \rightarrow [0, 1]$ (The data provider's trust threshold for performing an operation on a data item)
$T\_U : U \times P_D \rightarrow [0, 1]$ (A data user's trust degree for performing an operation on a data item)
$F : S \times U \times P_D \rightarrow \{0, 1\}$ (Trust-based decision)

In a trust-based decision, $F : S \times U \times P_D \rightarrow \{0, 1\}$ denotes a mapping from the data user's operation permission on the data item to the set $\{0, 1\}$. Here 1 denotes that access is permitted and 0 denotes that access is denied.

When a data user requests to perform an operation on a data item, the access control system judges whether the trust degree of the data user is higher than the data provider's trust threshold or not, and then decides to map the access permission to 0 or 1. That is,

$$\forall s \in S, u \in U, p_d \in P_D$$
$$F(s, u, p_d) = T\_U(u, p_d) \geq TT\_S(s, p_d)$$

If the trust degree of data user u for performing operation on d is not less than the data provider's trust threshold, the access permission is mapped to 1 and access is permitted; otherwise, access permission is mapped to 0 and access is denied. This can be seen as an instance of the trust enhanced security model and framework recently proposed in [12].

4.2 Ongoing access control mechanism

The above trust-based decision mechanism handles access security before access, but does not consider the authorization of data provider or data items' security sensitivity during the data usage. In the process of access control management, the ongoing access control mechanism is needed in order to achieve an efficient access control management.

As far as an authorization is concerned, the first step is to find all valid authorizations under the request. This is checked by the valid authorization function.

Authorization check function The valid authorization function is used to judge whether the current authorization *au* is valid. It can be expressed as follows:

$$G(r) = \begin{cases} au \text{ if } (r(u) = au(u)) \wedge (r(d) = au(d)) \wedge (r(priv) = au(priv)) \\ \qquad \wedge (r(p) = au(p)) \wedge (r(t) \subseteq au(t)) \\ \\ \phi \quad \text{others} \end{cases}$$

Here $au \in AU$, and $G(r)$ returns a set of valid authorizations. Except checking for the same data user to perform the same privileges on the same data items for the same purpose, the period constraint of an authorization plays an important role. If the request access time is within the retention period, it refers to the authorization is valid, otherwise, the authorization is invalid.

However, a valid authorization function is not enough for an access request, since it only checks whether an authorization exists in the current AU from the angle of the retention period. Besides that, the generalization level decides whether the access of the request is valid according to the current authorizations. Therefore, a valid access function is needed conveniently. Here, we use $r(gl)$ to indicate the generalization level that the request is going to access. If there exists a valid authorization satisfying $r(gl) = au(gl)$ (where $au(gl)$ refers to the generalization level in this authorization) the access is permitted, otherwise, the access is rejected.

Access check function The valid access function can be expressed as follows:

$$F(r) = \begin{cases} true \quad \exists au \in G(r), r(gl) = au(gl) \\ \\ false \text{ others} \end{cases}$$

where r is an access request. If $F(r)$ is true, the access is valid. Otherwise, it is invalid. After a data user submits an access request r and $F(r)$ is true, the user is permitted to access the data.

4.3 Process of access control management

In our privacy-aware access control model, the most important thing is that all authorizations are derived from the permissible usage of each data item. By specifying the retention period and purposes, the data items can only be accessed for the specific purpose during the valid period of usage, while, by applying data generalization techniques, authorizations for a user to access data items in specific generalization boundaries are specified. Three different attributes are required to meet these authorizations:

- *The time interval.* This includes the start time and end time for which access is permitted. At the end time, the privilege for using data items is revoked.
- *The access purpose.* Access to a data item can be permitted only for specific purpose.
- *Generalization level.* The data item can only be accessed under the authorized generalization level.

If a requested authorization tuple is a time independent authorization, then the authorization *au* is invoked. If it is temporal authorization, when the time exceeds the retention time, the *au* is illegal. If the data item being accessed is not in the same generalization level, access is rejected. The pseudo code of the ongoing access control policy is described in Algorithm 1.

```
Algorithm 1: Access control($AU$, $r$)
Input: an access request $r$ and the set of current
   temporal generalized authorizations $AU$
  Let $G(r) = \{au|au \in AU\}$; /*use the valid authorization function to return a set
                    of authorization tuples, and then judge whether the
                    authorization is valid*/
  If $G(r) == \phi$
      return false; /*This authorization does not exist*/
  else if $r(t) \nsubseteq au(t), \forall au \in G(r)$
      return false; /*No legal Authorization*/;
  else if
      let $k = F(r)$; /*use the valid access function to return a boolean value,
                   with which to judge whether the access is valid.*/
          if $r(gl) \neq au(gl), \forall au \in G(r)$
            then $k == false$
              return false; /*The access is rejected'*/
  else
      $k == true$
      return true. /*The access is succeeded'*/
```

5 State transitions

In previous sections, the privacy-aware access control model has been discussed in detail. In this section, the state transition of the proposed access control model is given.

Trust evaluation, authorization function and access function are decision factors employed by our privacy-aware access control model to determine whether a requester should be allowed to access an object. In addition to these factors, modern information system requires another important properties called ‘mutability’. Mutability means that the requester’ trust values and data attribute values can be updated as side-effects of access actions. When a query arises from an access requester, trust calculator calculates the trust value for a requester based on both observed history and records in recommendation databases. A trust value is passed to an access control manager for decision. The access control manager looks up access control policies that include pre-access and ongoing access mechanisms. The constraint service module evaluates access control constraints, for example, time, location, and memberships. The architecture outlined in Figure 3 provides the process of access control management.

Re-calculation and re-evaluation systems may cause the revocation of current enrollment or on-going access. Reports about the misbehavior of a requester will be sent to re-calculation system. The negative report may include ignorance of obligation, dishonest behaviors, or the revocation of a requesterąŕs certificate. When the trust value of the request drops below a minimum threshold, the on-going granted privilege will be revoked. The execution of the request is canceled. The attribute mutability of the principal, objects, or a context will be sent to re-evaluation system after the permission is granted. Once the system receives an event, the corresponding access control polices are rechecked if necessary (e.g., to allow an on-going usage to continue or revoke it). The re-calculation system tests whether the behaviors of the requester are too malicious to tolerate, while, the re-evaluation system checks whether the requester is violating the access control rules. Therefore, either one of two trigger events will result the revocation of the in-progress permission.

Further, an administrator of the system can make a forced revocation decision. For example, if a security administrator notices that a data user often sends many

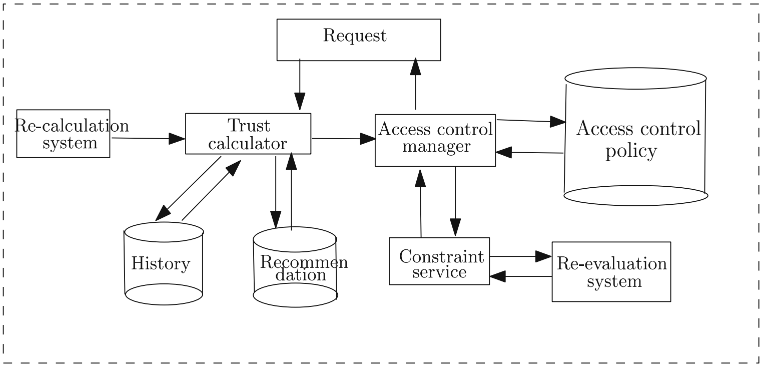

Figure 3 Process of access control management.

access requests without using services, the administrator may take actions on this user, such as revoking his authorization to prevent denial of service (DoS) attacks.

In the practical access control process, authorizations are assumed to be done before access is allowed (pre-check). However, it is quite reasonable to extend this for continuous enforcement by evaluating usage requirements throughout usage (on-going). The presence of on-going decisions is called the continuity. In the pre-access stage, we need to check whether the requester's trust degree is higher than the data provider's trust threshold and the required obligations and conditions are satisfied. In the ongoing-access stage, we need to check whether the valid authorization and access functions are satisfied. The on-going access may be revoked if the security policies are not satisfied. The pre-access decision policy and ongoing access control policy combined together construct the secure protection system. The state transition of privacy-aware access control actions is given in Figure 4. The states and actions in Figure 4 are explained below.

(1) Initial: the initial state of the metadata.
(2) Data generalization: replacing a data value with a less specific but semantically consistent value.
(3) Generalization boundaries: restricting the maximum allowed generalization level of each data item.
(4) Permissible usage: the type of potential data usage (i.e., purpose).
(5) Authorization: granting privileges of service to data users if data users meet authorization requirements of the system.
(6) Access request: a user request to access digital objects.
(7) Trust evaluation: checking whether the requester is trustworthy or not.
(8) Pre-access check: checking whether the trust threshold is satisfied.
(9) Permitted and denied: if the requester is trustworthy, the access to data items is permitted; otherwise, denied.
(10) Valid authorization check: checking whether the requested access time is in the valid retention period.
(11) Continued and revoked: if the time interval is not expired during the valid period, an access to data items is continued; otherwise, it is revoked.
(12) Valid access check: checking the accessed generalization level of the data item.
(13) Revoke privilege and endaccess: if the data item is accessed in a wrong generalization level, the system will revoke the privileges.

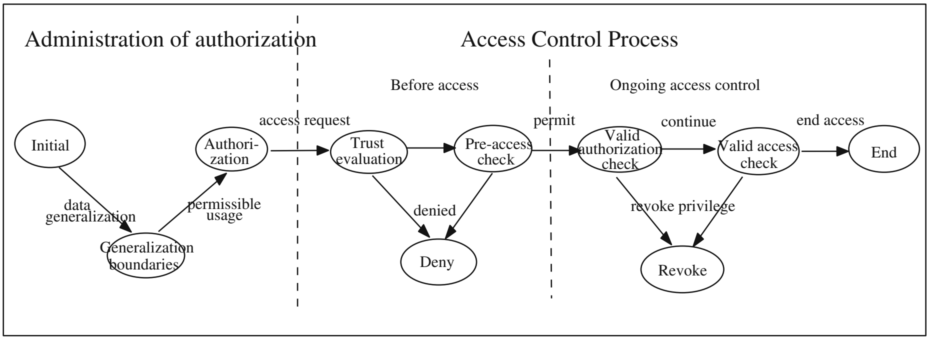

Figure 4 The state transition of privacy-aware access control model.

(14) Deny, Revoke and End: three final states. Deny is the state of refusing to access without revoking privileges. Revoke is the state after the action of revoke privileges, while End is the state after the action of endaccess.

From the analysis of state transitions in a privacy-aware access control, it is clear that an access is not a simple action, but consists of a sequence of actions and active tasks.

6 Experimental evaluations

The main goals of the experiments are two-fold. First, we study the performance and storage overheads of our proposed access control model. We consider the impact of the number of attributes accessed and the number of generalization levels on the execution time and storage overheads. We also examine the scalability of our approach by experimenting with relations of different cardinalities. Second, we investigate the effectiveness of our model in terms of *disclosure rate*, which is a novel metric defined to measure to what extent the access control models can protect the sensitive information from being discovered.

6.1 Experimental setup

We employ two data sets in our experimental evaluations. One is the real-life CENSUS data set, downloadable at http://www.ipums.org, and the other one is the synthetic numeric data set with the values between 0 and 100. To evaluate the efficiency and storage overheads, we adopt a real-world data set CENSUS, which contains the personal information of 500K American adults. The data set has nine discrete attributes summarized in Table 6. From CENSUS, we create two sets of micro tables, in order to examine the influence of dimensionality and the impact of cardinality. The first set has six tables, denoted as CENSUS_10K, $\cdots$, CENSUS_60K, respectively. Specifically, CENSUS_n $(10K \leq n \leq 60K)$ indicates the data set consisting of n records randomly sampled from the whole CENSUS data set, and each record has nine attributes shown in Table 6. The second set contains seven tables, denoted as 3-CENSUS, $\cdots$, 9-CENSUS, respectively, where n-CENSUS $(3 \leq n \leq 9)$ represents the data set with the first n attributes selected from Table 6, and each data set has the same number of records with the whole CENSUS data set. We evaluate the execution time of our approach by varying the cardinality

Table 6 Summary of attributes in CENSUS.

Attribute	Number of distinct values
Age	78
Gender	2
Education	17
Marital	6
Race	9
Work-class	8
Country	83
Occupation	50
Salary-class	50

of the data sets, the number of attributes and the number of generalization levels. We adopt the peak memory to measure the storage overheads, which indicates the maximum memory used during the implementation.

To evaluate the effectiveness of our proposed access control model, we generate a synthetic data set with 50K records, and each record contains 1,000 numeric attributes with the values randomly chosen from [0, 100]. Without loss of generality, in this set of experiment, we set the number of generalization levels to be three, High(H), Medium(M) and Low(L), where L level has the original specific value, and M level contains two intervals, [0, 50] and (50, 100], if the value at the L value is within [0, 50], then after the one level generalization, it becomes the interval [0, 50], otherwise it will be (50, 100]. H level specifies the most general information, which is the interval [0, 100]. For example, the number 80 is at L level, the interval (50, 100] is at the M level after the first generalization, and [0, 100] is at the H level generalization. We vary the portion of the attributes with different access levels and investigate their impact on the measurement of *disclosure rate*. In order to reduce the randomness, we run the each test for 500 times for each data and use the average to mark the graph.

6.2 Efficiency and effectiveness

In the following parts of the paper, we describe and explain the experimental results of the implementation of our proposed privacy-aware access control model in terms of its efficiency and effectiveness.

Generalization hierarchy In this set of experiments, the generalization hierarchy is generated as follows. Let D be a numeric data set with n attributes $(A_1, \cdots, A_n)$ and m records $(D_1, \cdots, D_m)$, and we are going to divide each attribute into l levels in its generalization hierarchy. For each attribute A_i, let $A_i(Min)$, $A_i(Max)$ be its minimum and maximum value and $A_i(j)$ be the jth values of the attribute A_i $(1 \leq i \leq n,\ 1 \leq j \leq m)$. After the p $(1 \leq p \leq l)$ levels generalization, if for some integer k $(1 \leq k \leq l - p + 1)$, the specific value $A_i(j) \in [A_i(Min) + \frac{A_i(Max)-A_i(Min)}{l-p} \times (k-1), A_i(Min) + \frac{A_i(Max)-A_i(Min)}{l-1} \times k]$ will become the interval $[A_i(Min) + \frac{A_i(Max)-A_i(Min)}{l-p} \times (k-1), A_i(Min) + \frac{A_i(Max)-A_i(Min)}{l-1} \times k]$. For example, we are going to find the four-level generalization hierarchy for some attribute $A_1 = \{1, 4.5, 7\}$. $A_1(Min) = 1$, $A_1(Max) = 7$, and for the second level generalization, there are three intervals generated, which are [1, 3], (3, 5] and (5, 7]. For the third level generalization, the formed intervals are [1, 4] and (4, 7]. Finally, the fourth level generalization interval is [1–7]. The generalization hierarchy for A_1 is shown in Figure 5.

Efficiency Figure 6 shows the comparison of running time when the data is generalized to different levels. In this experiment, we set the number of generalization level to be 6, which means the generalization hierarchy of each attribute has six levels and we investigate the execution time that the data are generalized to the ith level $(1 \leq i \leq 6)$ by varying the data percentage using data sets CENSUS_10K, $\cdots$, CENSUS_60K, and by varying the number of attributes using data sets 3-CENSUS, $\cdots$, 9-CENSUS. We notice that there is a sudden drop in Figure 6a when generalized to from the fifth level to the sixth level. This phenomenon happens on all six data

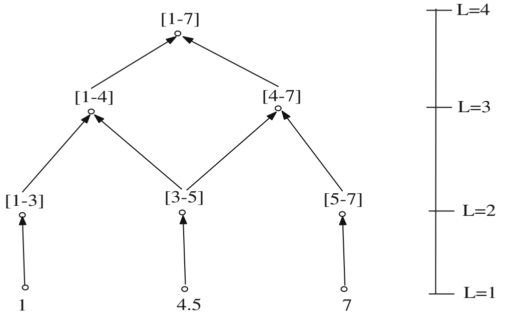

Figure 5 Example of generalization hierarchy for the numeric data set.

sets, CENSUS_10K, $\cdots$, CENSUS_60K (shown in Figure 6a), and this is because for each attribute, when the data is generalized to the highest level, all the data will be the same interval whose two interval points are the minimum and maximum values of the attribute, and this generalization obviously incurs less cost. The similar trend appears in Figure 6b as well when varying the number of attributes, and it can be explained similarly.

Figure 7a and b show the computation overhead of our proposed privacy aware access control model with generalization boundaries. In this set of experiments, the computation is run through six data sets CENSUS_10K, $\cdots$, CENSUS_60K, and the default number of attributes is 9 and the generalization hierarchy is set to have three levels. As shown in Figure 7a, the computation overhead increases as the number of records grows. As expected, the running time performance becomes poorer as the cardinality of the data set increases. Figure 7b plots the effect of the number of attributes on the execution time. The result is expected since the cost of computing is increased with the more dimensions. Figure 7c describes how the number of generalization levels affects on the computation overhead. From the figure, we can see that the running time is almost steady while varying the number of levels in the generalization hierarchy.

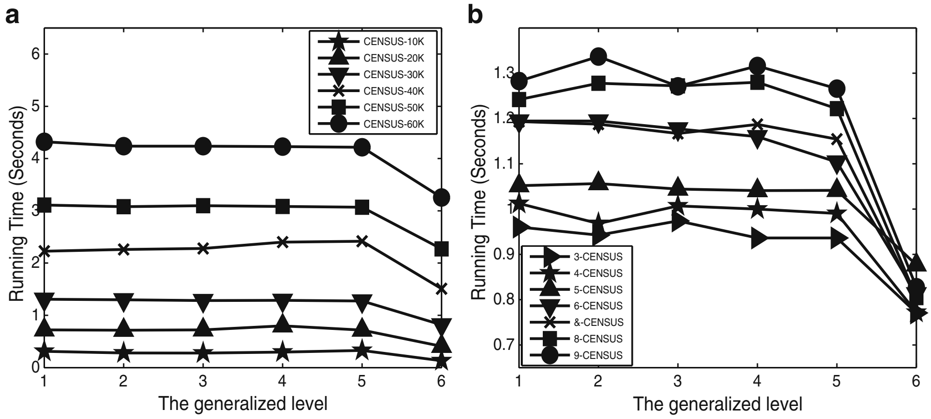

Figure 6 Computation overhead comparison when generalized to the specified level vs. **a** the data percentage varies; **b** the number of attributes varies.

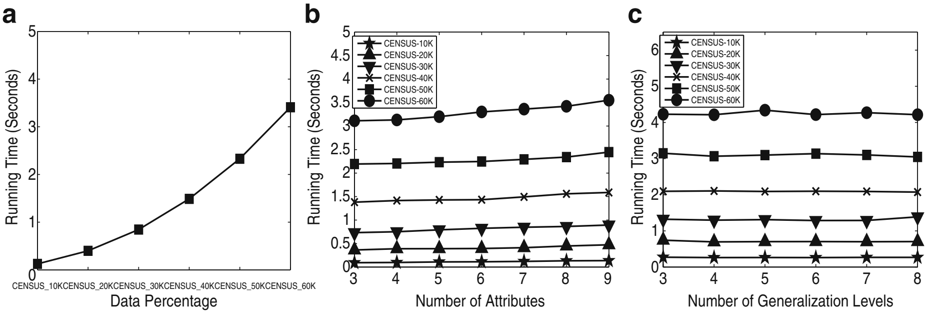

Figure 7 Computation overhead comparison when **a** the data percentage varies; **b** the number of attributes varies; **c** the number of generalization levels varies.

Storage overhead Figure 8a and b display the space overhead of our proposed access control model. As shown in Figure 8a and b, the storage overhead increases when the number of records grows and the number of accessed attributes increases. This is because more data records or more data dimensions lead to the higher volume of memory consumed. Figure 8c shows the memory usage when varying the number of generalization levels. From the graph, the more levels are divided in each generalization hierarchy, the more memory is needed to store them, since a larger number of levels leads to the more fine-grained the information on each level, which results in higher memory usage.

Effectiveness Having verifying the efficiency of our technique, we proceed to test its effectiveness. In this set of experiments, we use the *disclosure rate* to measure the effectiveness of our proposed access control model with generalization boundaries. We are going to use H, M and L to denote the High, Medium and Low level in the classification of the generalization boundaries, respectively. Recall that in our privacy-aware access control model, if a data user issues an access request, the access to each attribute is specified with generalization boundaries. Suppose there are n attributes in the database, among which there are n_H attributes which are generalized to H level, n_M attributes are generalized to M level, and n_L attributes

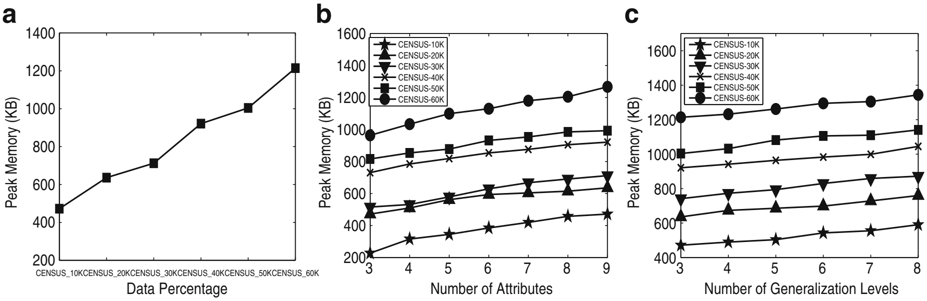

Figure 8 Storage overhead comparison when **a** the data percentage varies; **b** the number of attributes varies; **c** the number of generalization levels varies.

are generalized to L level, where $n_H + n_M + n_L = n$. In this case, the requester could totally access information in $n_H + 2n_M + 3n_L$ levels, which indicates the number of secure access. Consider the situation where there is no specification of generalization boundaries, for each attribute, the data user could access any three-level information. Then there would be $3(n_H + n_M + n_L)$ access, and among those, there will be $3(n_H + n_M + n_L) - (n_H + 2n_M + 3n_L)$ insecure access. Thus, we define the *disclosure rate* as $1 - \frac{n_H+2n_M+3n_L}{3(n_H+n_M+n_L)}$. The lower the rate is, the more secure the access control model would be.

The results are shown in Figure 9. Figure 9a displays the disclosure rate by varying the portion of L from 0 to 100%, and the portion of the other two levels are randomly generated. From the graph, we can see that the disclosure rate is decreasing as the amount of L increases. This is easy to understand, since the more the L level are specified in the generalization boundary, the less the insecure access are and the lower the disclosure rate is. Figure 9b describes the disclosure rate by varying M from 0 to 100%, and the portion of the other two levels are randomly generated. The graph shows that the disclosure rate almost remains unchanged with the increased portion of M. Figure 9c reports the effect of H on the disclosure rate. When varying the portion of H from 0 to 100%, and the portion of the other two levels are randomly generated. The disclosure rate is ascending. It indicates that the more H level attributes are specified in the generalization boundary, the more information would be disclosed in traditional access control model, which demonstrate our proposed access model could better avoid the information disclosure by specifying generalization boundaries. Therefore, in this case, our privacy-aware access model is superior to the traditional access control model.

6.3 Experiment summary

In the series of experimental studies, we implemented our proposed privacy-aware access control model with generalization boundaries, and evaluated its efficiency and effectiveness. We measure the efficiency in terms of the time complexity and its storage overhead, and quantify the effectiveness by using a new metric called "disclosure rate", which reflects to what extent the access control model can protect the sensitive information from being revealed.

We use a real-life data set CENSUS in implementing our proposed model for verifying its efficiency. We evaluate the time and space complexity by varying both

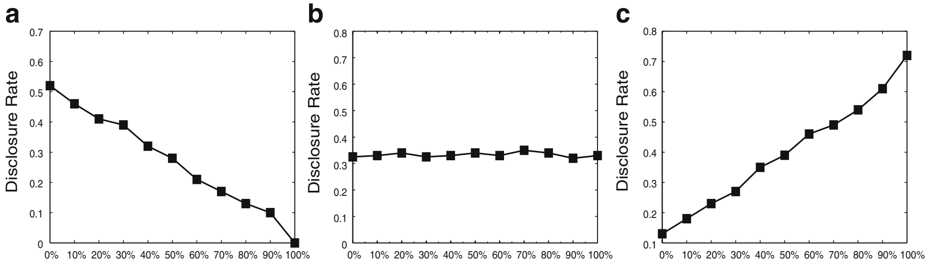

Figure 9 Disclosure rate comparison when varying **a** the number of L levels; **b** the number of M levels; **c** the number of H levels.

the data percentage, the number of attributes and the number of generalization levels, in addition, by setting a specified maximum generalization level l, we also evaluate the efficiency by generalizing the data from different levels to l. The proof-of-concept experiments support that by using the generalization boundary technique, our proposed privacy-aware access control model is practical.

The effectiveness studies are carried on a synthetic data set with numeric values. We defined the generalization rule to guide the data generalization, and without the loss of generality, we experiment the model by allowing three privacy levels, H, M and L representing *High*, *Medium* and *Low*. We introduce a new measurement *disclosure rate* to quantify the insecure accesses and its portion among all the valid accesses. By comparing with the traditional access control model, our proposed privacy-aware access control model with generalization boundary has been proved to be useful.

7 Related work

To date, several privacy protecting access control models have been proposed to deal with various aspects of the problem of high-assurance privacy systems [1, 3, 4, 11].

The W3Cs Platform for Privacy Preference (P3P) [25] allows web sites to encode their privacy practice, such as what information is collected, who can access the data for what purposes, and how long the data will be stored by the sites, in a machine-readable format. P3P enabled browsers can read this privacy policy automatically and compare it to the consumer's set of privacy preferences that are specified in a privacy preference language such as a P3P preference exchange language (APPEL) [26], also designed by the W3C. Even though P3P provides a standard means for enterprises to make privacy promises to their users, P3P does not provide any mechanism to ensure that these promises are consistent with the internal data processing. By contrast, the work in our paper not only provides an effective generalization strategy to maximize data privacy and usability but also provides details on how to manage the valid access process. In particular, we propose a privacy-aware access control model based on the generalization techniques.

The concept of Hippocratic databases that incorporates privacy protection within relational database systems was introduced by Agrawal et al. [3]. The proposed architecture uses privacy metadata, which consists of privacy policies and privacy authorizations stored in two tables. Byun et al. [6, 7] presented a comprehensive approach for privacy preserving access control based on the notion of purpose. In the model, purpose information associated with a given data element specifies the intended use of the data element, and the model allows multiple purposes to be associated with each data element. The granularity of data labeling is discussed in detail in [6], and a systematic approach to implement the notion of access purposes, using roles and role-attributes is presented in [7]. Although these models do protect the privacy of data providers, they are rigid and do not provide ways to maximize the utilization of private information. More specifically, in these models, the access decision is always binary; i.e., a data access is either allowed or denied as in most conventional access control models. Different from previous models, the novelty of our approach is that our model can provide a much finer level of access control as the access decision is based on the question of "how much information can be allowed for a certain user", rather than "is information allowed for a certain user or not". In other words, every piece of information is classified into different generalization levels and every user is assigned an authorization to access the private information.

Previous work on multilevel secure relational databases [10, 14] also provides many valuable insights for designing a fine-grained secure data model. In a multilevel relational database system, every piece of information is classified into a security level, and every user is assigned a security clearance. Based on this access class, the system ensures that each user gains access to only the data for which s/he has proper clearance, according to the basic restrictions. Byun and Bertino [5] proposed a new class of access control systems based on the notion of *micro-view*, which applied the idea of views at the level of the atomic components of tuples to an attribute value. However, the model in [5] is not a complete solution but rather it is aimed to show some of the capabilities. Some technical challenges raised by their model have been solved in our paper. One of the challenges is to balance the trade-off between data privacy and data usability. We solve this challenge by introducing the privacy-aware generalization boundary technique, which can maximize the privacy and utility for both data providers and data users. Another challenge is concerned with the applicability to general-purpose access control, which we solve by providing a complete access control model with the implementation of access control policy. We also discuss the state transition and architecture of our privacy-aware access control model.

8 Conclusions and future work

In this paper, we have considered a generalization boundary technique that can satisfy the requirements of both data providers and data users. Both privacy and usability of data items can be achieved when the data item is generalized using this technique. Moreover, we present a privacy-aware access control model, where the trust-based decision policy and ongoing access control policy combine together to create a secure protection system. Further, our model provides a much finer level of control as the access control decision is based on the question of "how much information can be allowed for a certain user", rather than "is information allowed for a certain user or not". The privacy-aware access control model presented in this paper provides an example of multi-level secure in relational databases.

Our proposed model provides efficient generalization strategies for privacy preserving access control systems, but much more work still remains to be done. The future work includes devising a high level language in which privacy specifications can be expressed precisely. We also plan to extend our model to cope with complex query processing. We will introduce the queries with join, sub-queries or aggregations into our model. These are challenging problems, but they are vital elements of a comprehensive privacy protection framework.

References

1. Adam, N.R., Worthmann, J.C.: Security-control methods for statistical databases: a comparative study. CSUR **21**(4), 515–556 (1989)

2. Agrawal, R., Evmievski, A., Srikant, R.: Information sharing across private databases. In: Proc. of the 2003 ACM SIGMOD Int. Conf. on Management of Data. ACM Press (2003)
3. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic databases. In: Proceedings of the 28th International Conference on Very Large Databases (VLDB) (2002)
4. Ashley, P., Powers, C.S., Schunter, M.: Privacy promises, access control, and privacy management. In: Third International Symposium on Electronic Commerce (2002)
5. Byun, J.W., Bertino, E.: Micro-views, or on how to protect privacy while enhancing data usability: concepts and challenges. SIGMOD Rec. **35**(1), 9–13 (2006)
6. Byun, J.W., Bertino, E., Li, N.: Purpose Based Access Control for Privacy Protection in Relational Database Systems. Technical Report 2004-52, Purdue University (2004)
7. Byun, J.W., Bertino, E., Li, N.: Purpose based access control of complex data for privacy protection. In: Symposium on Access Control Model And Technologies (SACMAT) (2005)
8. Dong, X., Madhavan, J., Nemes, E.: Reference reconciliation in complex information spaces. In: ACM International Conference on Management of Data (SIGMOD) (2005)
9. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed nist standard for role-based access control. ACM Trans. Inf. Syst. Secur. **4**(3), 224–274 (2001)
10. Jajodia, S., Sandhu, R.: Toward a multilevel secure relational data model. In: ACM International Conference on Management of Data (SIGMOD), pp. 50–59. ACM Press, New York (1991)
11. LeFevre, K., Agrawal, R., Ercegovac, V., Ramakrishnan, R., Xu, Y., DeWitt, D.: Disclosure in hippocratic databases. In: The 30th International Conference on Very Large Databases (VLDB) (2004)
12. Lin, C., Varadharajan, V.: Trust enhanced security for mobile agents. In: Proc of the 7th IEEE International Conference on E-Commerce Technology, CEC 2005, Germany, July 2005. ISBN 0-7695-2277-7; ISSN 1530-1354 (2005)
13. Sandhu, R.: Role hierarchies and constraints for lattice-based access controls. In: European Symposium on Research in Security and Privacy (1996)
14. Sandhu, R., Chen, F.: The multilevel relational data model. ACM Trans. Inf. Syst. Secur. **1**(1), 93–132 (1998)
15. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role based access control models. IEEE Computer **29**(2), 38–47 (1996)
16. Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: ACM International conference on Knowledge discovery and data mining (SIGKDD) (2002)
17. Seamons, K., Winslett, M., Yu, T.: Limiting the disclosure of access control policies during automated trust negotiation. In: Proc. of NDSS'01, pp. 109–125. IEEE Press (2001)
18. Sun, X., Wang, H., Li, J., Truta, T.M.: Enhanced P-sensitive K-anonymity models for privacy preserving data publishing. Transactions on Data Privacy (TDP) **1**(2), 53–66 (2008)
19. Sun, X., Wang, H., Li, J.: L-diversity based dynamic update for large time-evolving microdata. Australasian Conference on Artificial Intelligence (AI) **2008**, 461–469 (2008)
20. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems (IJUFKS) **10**(5), 571–588 (2002)
21. Tumer, A., Dogac, A., Toroslu, H.: A semantic based privacy framework for web services. In: Proc. of ESSW'03 (2003)
22. Wang, Y., Vassileva, J.: Trust and reputation model in collaborative networks. In: Proc. 3rd IEEE Int. Conf. Collaborative Computing, pp. 150–157 (2003)
23. Westin, A.: E-Commerce and Privacy: What Net Users Want. Technical Report, Louis Harris & Associates (1998)
24. Westin, A.: Freebies and Privacy: What Net Users Think. Technical Report, Opinion Research Corporation (1999)
25. World Wide Web Consortium (W3C). A P3P Preference Exchange Language 1.0 (APPEL 1.0). Available at www.w3.org/TR/P3P-preferences
26. World Wide Web Consortium (W3C). Platform for Privacy Preferences (P3P). Available at www.w3.org/P3P