A Practical Database Security Model Using Purpose-Based Database Access Control and Group Concept

Ji-Young Lim
Department of Computer Science
Yonsei University
Seoul, Korea
jylim@cs.yonsei.ac.kr

Woo-Cheol Kim
Department of Computer Science
Yonsei University
Seoul, Korea
twelvepp@cs.yonsei.ac.kr

Hongchan Roh
Department of Computer Science
Yonsei University
Seoul, Korea
fallsmal@cs.yonsei.ac.kr

Sanghyun Park
Department of Computer Science
Yonsei University
Seoul, Korea
sanghyun@cs.yonsei.ac.kr

Abstract— **Personal information that is collected online can be misused and abused. Therefore, data security techniques that restrict the usage of data only to purposes specified by data providers are needed. The LDHD model, a well-known database security model, expresses the purpose of data provision in the unit of cell in order to have a detailed control over privacy preservation. However, since this model collects metadata for every pair of users and purposes, its metadata becomes much larger than the actual data themselves and the introduction of a new data user incurs significant changes to the metadata. Thus, it is just an ideal database security model which can hardly be applied to current database management systems. To resolve these problems, this paper first identifies the requirements of database management systems supporting privacy preservation and then suggests a practical database security model called PBDM+G. Instead of collecting metadata for every pair of users and purposes, the PBDM+G model collects it for every purpose, and the collected metadata are grouped for duplicate elimination. The experimental result shows that the PBDM+G model consumes at most 10% of the space needed for the LDHD model while reducing query processing time up to 23.6%.**

Keywords-database security; access control; privacy preservation

I. INTRODUCTION

With increases in the number of internet users and the types of online activities, such as internet banking and e-commerce, more and more personal information is being collected and transferred via the internet. However, personal information is generally gathered without the knowledge or permission of data providers. Thus, despite the legality of a company's information gathering practices, the privacy of online service users is often compromised. Many researches have reported privacy infringements caused by excessive data collection, and the reasons for privacy preservation have been thoroughly presented [1–4]. In a survey commissioned by the National Consumers League and Harris International [5], about 56% of respondents showed concern regarding their privacy and security.

As a result, companies have introduced methods for preserving personal privacy, such as providing online documentation of their privacy policy and inducing online privacy seal programs. However, consumers are still concerned about the manner in which their information is used and distributed by collectors, perhaps because existing solutions to privacy infringement largely focus on the post-censorship of personal information misuse and abuse. For the sake of the fundamental preservation of personal privacy, personal information must be collected together with the purposes of data usage, mostly specified by data providers, and must only be used for purposes explicitly specified by the user.

Existing database security systems have focused on preventing the spread of collected information to risky users, primarily by allowing only authorized data users to access the underlying databases. Typical examples using this concept include Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC) [6]. These traditional database models preserve previously collected data, but they cannot ensure that data are always used as intended by data providers.

Recently, Agrawal et al. [7] proposed the Hippocratic database model, a new database model for privacy preservation. Unlike traditional models, this model collects metadata during the data gathering process and uses the metadata to preserve privacy. Since this model is based on record-level metadata collection, it has such drawbacks as the limitation of data collection and the possibility of data leakage when the privacy policy is changed. To overcome such drawbacks, LeFevre et al. [8] proposed the LDHD (Limiting Disclosure in Hippocratic Databases) model, a new privacy preservation model based on cell-level metadata collection. However, the LDHD model incurs large storage overhead for metadata and thus is inappropriate for large databases. Furthermore, the introduction of each new data user incurs significant changes to the metadata. Thus, it is just an ideal database security model which can hardly be applied to current database management systems.

To solve the problems of the LDHD model, this paper suggests a practical database security model which is equipped with a couple of unique features as well as cell-level metadata management.

II. REQUIREMENTS OF DATABASE SYSTEMS SUPPORTING PRIVACY PRESERVATION

Fundamental to personal privacy preservation is the ability of data providers to personally control access to their information, which can be achieved only when the design of every data gathering and processing stage considers privacy preservation. In this section, for each data collection and processing stage, we identify what is required of database systems to ensure privacy preservation

A. Data collection

Let us first consider the data collection stage. This stage has not been heavily emphasized in traditional database security

models, but the database security models designed for privacy preservation do emphasize the significance of this stage.

The most crucial requirement of the data collection stage is that not only the personal information of data providers but also the permissible purposes of data usage must be collected (*requirement 1*). Secondly, all personal information presented with the purposes of data usage must be collected (*requirement 2*). Thirdly, the data protection unit must be as small as possible to ensure detailed privacy control (*requirement 3*). More specifically, cell-level protection is more desirable than record-level or table-level protection. Finally, the process for data providers to specify the purposes of data usage must be simple (*requirement 4*). For example, choosing from a set of pre-defined usage purpose patterns may be easier than specifying usage purposes one by one.

B. *Data store and management*

As mentioned, with the aim of fundamental preservation of personal privacy, the metadata of usage purposes must be collected with personal information. However, since most commercial database management systems are optimized to handle a large amount of 'actual' data, they are limited in their ability to maintain a large volume of metadata. Thus, it is essential that database systems supporting privacy preservation strive to reduce the size of metadata by discovering the regularities hidden in them (*requirement 5*). In addition, the introduction of new data users or new usage purposes must not significantly alter the metadata (*requirement 6*).

C. *Query processing*

Most commercial database systems take a two-step approach to processing queries from data users. Access authorities are checked in the first step, and requested data are located and retrieved in the second step. The second step is performed only when the data user issuing a query is determined to have permission to access the target data. Database systems designed for privacy preservation need an extra step between the first and second steps where the usage purposes of data users are compared with those of data providers. For such an extra step, the systems must provide either a new SQL construct or a query rewrite module for appending a 'check' clause to an original user query (*requirement 7*). In addition, the systems must be able to cope with the situations where data users have more than a single usage purpose (*requirement 8*).

III. TRADITIONAL DATABASE SECURITY MODELS

The policies for controlling database access authority can be classified in three ways [6]. With DAC (Discretionary Access Control), where the owner of a particular data item determines other users' access authorities for his data item. With MAC (Mandatory Access Control), access authorities are determined and controlled by the security levels of data items and the authority levels of data users. Lastly, with RBAC (Role-Based Access Control), access authorities are assigned to roles to which data users belong.

The policies mentioned above may be implemented by employing data structures such as access control matrices [9], access control lists [10], and capability lists [11]. Access control matrices are used to rapidly examine the access authorities of a given data user and data item pair. Access control lists are used to effectively identify all data items that can be accessed by a specific data user. Capability lists are used to expedite the process of identifying all data users with the authorities to access a particular data item.

A. *Discretionary access control policy*

This policy controls access to data items by examining the identifiers of data users or user groups. Since data users holding the access authorities to a particular data item can grant their authorities to other data users, this policy is regarded as a self-control policy. Typical examples of this policy include task-grant model [13] and Wood model [14].

B. *Mandatory access control policy*

This policy allows a data user to access a certain data item only when his authority level matches the security level of the data item. Typical examples of this policy include Bell-Lapadula model [15] and Biba model [16].

C. *Role-based access control policy*

This policy determines the access authorities of data users according to the roles to which they belong, and thus prevents data users from accessing data items at their own will[17]. That is, in this model, access authorities are given to roles rather than data users, and, to access a target data item, data users must be a member of the roles possessing an appropriate authority. Data users become a member of a specific role according to their responsibilities and authorities. In addition, they are able to change their roles easily without requiring the modification of access structures.

IV. TRADITIONAL SECURITY MODELS SUPPORTING PRIVACY PRESERVATION

For a simpler explanation, the database environment is restricted as follows: there is only one data table in the database and each data record expresses the personal information of a single data provider. Such a restricted database environment can be easily generalized to a real database environment.

A. *Hippocratic database model*

The Hippocratic database model supports privacy preservation by utilizing metadata such as the 'privacy policy' and 'privacy authority'. The privacy policy, a database administrator's policy for data collection, describes the usage purposes of each attribute of the data table. The privacy authority, a policy for controlling access to data items, describes data users' privileges to access each attribute of the data table. By keeping metadata of privacy policy and privacy authority, the Hippocratic database model forces data items to be collected only for suitable usage purposes and accessed only by permitted data users.

For a given usage purpose, the 'privacy preference' of a data provider indicates whether to allow his or her data to be used for that purpose. In the Hippocratic database model, the personal data of a data provider can be collected only when their privacy preference matches the privacy policy of the database system. Therefore, the Hippocratic database model limits data collection and thus does not satisfy *requirement 2*. Furthermore, since it is based on record-level data protection, the Hippocratic model does not facilitate detailed privacy control and thus does not satisfy *requirement 3*.

B. *Limiting disclosure in Hippocratic databases*

To overcome the limitations of the Hippocratic database model, LeFevre et al. [8] suggested the LDHD (Limiting Disclosure in Hippocratic Databases) model, a new type of database security model based on cell-level data protection[18].

The data collection process of the LDHD model is as follows: First, when a data provider presents his data, he specifies his preference for every pair of usage purposes and data users registered in the system. As an illustration, let us suppose that the system has two data users (namely *A* and *B*) with usage purposes P_1 and P_2, respectively. Then, data providers have to indicate whether they are willing to release their data to (A-P_1) and (B-P_2). Such preferences are expressed as a binary format (that is, 1 and 0 for denoting approval and disapproval, respectively) and stored in a metadata table. The metadata table which stores data providers' preferences for every pair of data users and usage purposes is called the PreferenceTable(User-Purpose).

When receiving a query from a data user, the LDHD model rewrites the query to pass through the PreferenceTable(User-Purpose) for privacy checking. More specifically, the LDHD model rewrites the query to include a 'case' statement that returns either a NULL value if the data provision is not permitted by the PreferenceTable(User-Purpose), or the actual data otherwise.

In real database environments, however, the LDHD model is problematic for several reasons. First, data providers must specify their preferences as many times as the number of data users multiplied by the number of usage purposes. For example, consider a data provider who is about to submit his 'address' information to the system in which there are three data users possessing two usage purposes each. Then, he has to indicate whether he is willing to provide his 'address' information for each of six combinations (i.e., $3 \times 2=6$). The time consumption and difficulty associated with making the potentially high number of choices inherent to the LDHD model precludes it from meeting *requirement 4*.

The second problem is that the LDHD model must maintain a large volume of metadata. As previously stated, in the LDHD model, a single data table cell may correspond to more than a single preference. Thus, the volume of metadata is liable to exceed that of the actual data, and *requirement 5* may therefore not be met.

The third and last problem is that the introduction of a new data user significantly alters the metadata. For example, suppose that the system has two users, namely *A* and *B*, with usage purposes P_1 and P_2, respectively. When a new user *C* enters into the system with purpose P_2, then the system must request all existing data providers to specify their preferences for this new pair (i.e., pair of P_2 and *C*) in order to fill up a new table PreferenceTable(P_2-C). Thus, the LDHD model does not satisfy the *requirement 6*.

The relative pros and cons of the Hippocratic database model and the LDHD model are compared, in terms of the aforementioned requirements, in Table 1.

TABLE I. COMPARISONS OF THE HIPPOCRATIC DATABASE MODEL AND THE LDHD MODEL IN TERMS OF THE REQUIREMENTS OF DATABASE SYSTEMS FOR PRIVACY PRESERVATION

No	Hippocratic database model	LDHD model
1	fully supported	fully supported
2	partially supported	fully supported
3	record level	cell level
4	convenient	inconvenient
5	little metadata	much metadata
6	little change required	much change required
7	not supported	fully supported
8	not supported	not supported

V. PURPOSE BASED DATABASE MODEL USING PRIVACY PREFERENCE GROUPS

A. Purpose based database model (PBDM)

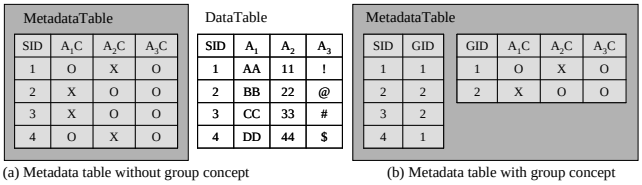
Careful consideration of the issues associated with privacy preservation aided in solving the problems of the LDHD model. Data providers are more likely concerned about the purposes for which their personal data are used than the data users by which their personal data are accessed. For instance, if a data provider provides his/her address to an online-shopping mall for a sole purpose of "delivery", he/she is more likely concerned if his/her address is used only for the purpose of delivery rather than concerned about the identities of the persons trying to access his/her data. Irrespective of the number of the users of his/her address data, he/she would feel safe if the address is accessed only for delivery. Based on this analysis, the PBDM (Purposed Based Data Model) is designed such that data providers express their preference for each usage purpose without considering data users.

B. Purpose based database model with grouping concept (PBDM+G)

Since the PBDM only collects privacy preferences for usage purposes, its metadata is much smaller than that of the LDHD model. Note, however, that the personal information of a data provider usually consists of multiple data items each of which corresponds to a single cell of a data table. Since the PBDM stores the privacy preference for every cell of a data table into its metadata, the size of the metadata still exceeds that of the actual data themselves.

However it is easily conceivable that numerous duplicates exist within the metadata of the PBDM. For example, let us consider a data table with three attributes and a single usage purpose P_1. Each data provider must choose either 'yes' or 'no' for each one of three attributes to indicate whether or not he is willing to allow his data to be used for P_1. Therefore, in this situation, there are eight distinct combinations of choices in total (i.e., $2^3=8$). So, if there are more than eight data providers in the system, there must be at least two data providers with the same privacy preferences on P_1. If we detect and remove such a duplicate within the metadata, then the volume of metadata relative to the number of users may be reduced.

To manage (i.e., detect and remove) such duplicates effectively, we propose to apply a *normalization* process to the tables maintained in the metadata. To do this, we first generate a set of privacy preference groups by aggregating data providers with the same privacy preferences. Then, rather than storing the privacy preference of each data provider individually, we store the privacy preferences of groups with the mapping information from data providers and apply it to privacy preference groups. This normalization process, which is illustrated in Fig. 1, results in the removal of metadata duplicates in the PBDM and generates a new database security model called the PBDM+G.



(a) Metadata table without group concept (b) Metadata table with group concept

Figure 1. Metadata tables before and after the grouping concept is applied.

C. Comparison of metadata schemas

Consider the data table shown in Fig. 2 which has a single key attribute *SID* and |A| non-key attributes. Traditional database systems that do not support privacy preservation just keep the usual schema information (e.g., information on a key attribute) in their meta tables, without maintaining extra metadata for privacy preservation. However, the LDHD model keeps additional meta tables `PreferenceTable(User-Purpose)`, each of which stores the preferences of data providers for a pair of data users and usage purposes. Note that the number of such tables equals the number of distinct pairs of data users and usage purposes (i.e., number of data users × number of usage purposes). Also, this model requires an additional meta table, the `UserPurposeMappingTable` to express the mapping information from data users to usage purposes. The data table and meta tables maintained in the LDHD model are illustrated in Fig. 2.

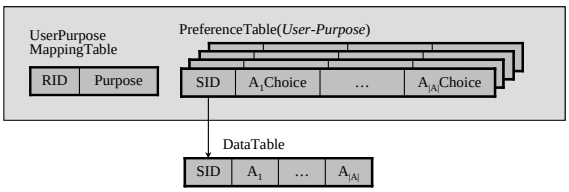


Figure 2. Metadata schema of the LDHD model

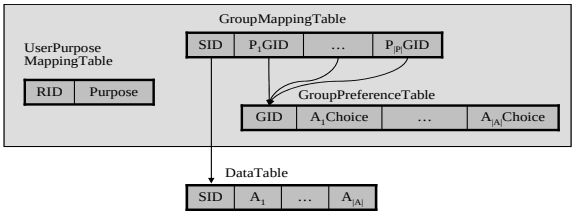


Figure 3. Metadata schema of the PBDM+G model

Let us now consider the meta tables of the PBDM and PBDM+G models. Just like the LDHD model, the PBDM model also needs the `UserPurposeMappingTable`, but it uses the meta table `PreferenceTable(Purpose)` rather than the meta table `PreferenceTable(User-Purpose)`. On the other hand, the PBDM+G model uniquely employs the `GroupPreferenceTable`. Instead of expressing the preference of each data provider, this table represents the preference of each privacy preference group. This model also employs the `GroupMappingTable`, which maps data providers and usage purposes to a corresponding group identifier (i.e., *GID*). Given a data provider and a usage purpose, we can extract his preferences for a specific set of attributes of the data table by joining the `GroupMappingTable` and `GroupPreferenceTable`. For example, given a data provider with 5 as his *SID*, to extract his preference for usage purpose P_2, we join the two meta tables with the following condition: `GroupMappingTable.`*SID* = 5 AND `GroupMappingTable.`P_2*GID* = `GroupPreferenceTable.`*GID*.

D. Comparison of metadata volume

To compute and compare the volumes of metadata of the aforementioned security models, we first define the following several variables: Let |S| and |A| denote the numbers of tuples and attributes of the data table, respectively. Also, let |P| and |R| be the number of usage purposes and number of data users, respectively. The metadata volumes of various security models may be approximated by their numbers of metadata cells, which may then be compared. The meta table `UserPurposeMappingTable` is excluded in this comparison because it is much smaller than the other meta tables and is contained in all of the security models compared herein.

Let us first consider the number of metadata cells of the LDHD model. The number of `PreferenceTable(User-Purpose)` tables equals the number of distinct pairs of data users and usage purposes. More specifically, if every data user has only a single usage purpose, then the LDHD model requires the smallest number of `PreferenceTable(User-Purpose)` tables, which is |R|. On the contrary, if every data user possesses all usage purposes, then the model needs the largest number of such tables, which is |R|×|P|. And, just like the data table, each one of `PreferenceTable(User-Purpose)` tables has (|A|+1)×|S| cells. Therefore, the total number of metadata cells of the LDHD model is between |R|×(|A|+1)×|S| and|P|×|R|×(|A|+1)×|S|.

Let us now consider the number of the metadata cells of the PBDM+G model. Since this model is based on the group concept, we need to employ one more variable |G|, which is the number of privacy preference groups generated through the normalization process. The PBDM+G model contains two meta tables, the `GroupMappingTable` and `GroupPreferenceTable`. The table `GroupMappingTable` requires|S|×(|P|+1) cells and the table `GroupPreferenceTable` needs |G|×(|A|+1) cells. Therefore, the total number of the metadata cells is expressed as (|S|×(|P|+1))+(|G|×(|A|+1)).

E. Query modification algorithm

When receiving a query from a data user, the LDHD model modifies it to include a 'case' statement for privacy checking. More precisely, a user's query is modified to provide actual data if the user is determined to have adequate permission but modified to return *null* otherwise. For instance, if data user *R* wants to retrieve two attributes, A_i and A_j, for the purpose of P_r, then his original query, "select A_i, A_j from DataTable", is modified as shown in Fig. 4.

```
select case when exists ( select  Ai
                from   PreferenceTableBy(R-Pr) as PT
                where  DT.SID = PT.SID and PT.AiChoice = 1 )
            then Ai else null end,
        case when exists ( select  Aj
                from PreferenceTableBy(R-Pr) as PT
                where  DT.SID = PT.SID and PT.AjChoice = 1 )
            then Aj else null end
from DataTable as DT
```

Figure 4. A query modified by the LDHD model; the original "select A_i, A_j from Data Table" is issued for the purpose of P_r by data user *R*

```
select  case when PT1.AiChoice = 0 then null else DT.Ai end,
     case when PT2.AjChoice = 0 then null else DT.Aj end
from   DataTable as DT left outer join GroupMappingTable as GMT
                on DT.SID = GMT.SID
            left outer join GroupPreferenceTable as PT1
                on GMT.PrGID = PT1.GID and PT1.AiChoice = 1
            left outer join GroupPreferenceTable as PT2
                on GMT.PrGID = PT2.GID and PT2.AjChoice = 1
```

Figure 5. A query modified by the PBDM+G model; the original query "select A_i, A_j from Data Table" is issued for purpose P_r by data user *R*

As described in Chapter 5, the PBDM+G model utilizes privacy preference groups and therefore its metadata schema is different from that of the LDHD model. The proposed PBDM+G model modifies the same query as above as shown in Fig. 5.

The relative pros and cons of the PBDM and PBDM+G database models are compared, in terms of the requirements of database systems for privacy prevention, in Table 2.

TABLE II. SUMMARY OF THE CHARACTERISTICS OF THE PBDM AND PBDM+G MODELS IN TERMS OF THE REQUIREMENTS OF DATABASE SYSTEMS FOR PRIVACY PRESERVATION

No	PBDM	PBDM+G
1	fully supported	fully supported
2	fully supported	fully supported
3	cell level	cell level
4	convenient	very convenient
5	much metadata	little metadata
6	little change required	little change required
7	fully supported	fully supported
8	not supported	not supported

VI. PERFORMANCE EVALUATION

A. Environments and settings for experiments

We compare the query processing times and metadata sizes of the four methods: 1) an original query processing method without consideration for privacy preservation, 2) the query processing method of the LDHD model, 3) the query processing method of the PBDM model, and 4) the query processing method of the PBDM+G model.

According to the experimental results [8], the LDHD model with the 'case' statement shows better performance than that with the 'outer join' statement. Therefore, only the LDHD model with the 'case' statement is compared with the other methods. Since the schema of the queries in the PBDM model is identical to that of the queries in the LDHD model, it is reasonable to assume that the processing times of both types of query are nearly identical. However, as their metadata schemas differ, a comparison of their metadata volumes remains meaningful.

For efficient query processing, we constructed indexes on meta tables as well as on the data table. In more detail, in addition to constructing the index on the primary key (i.e., column *SID*) of the data table, we built the index on the entire columns of table PreferenceTable(User-Purpose) for the LDHD model just like the experiments performed in [8]. And, for the PBDM+G model, we built the index on the primary key (i.e., column *GID*) of table GroupPreferenceTable and the index on the entire columns of table GroupMappingTable.

We measured the query processing time of each method 10 times and computed their average from the middle six values. We calculated the metadata volume of each method by adding the sizes of its meta tables and associated indexes. We uses the Wisconsin benchmark data set [19]. This data set was also used for experiments in [8].

The hardware platform for the experiments was Pentium IV PC equipped with 2.60GHz CPU and 512MB memory. And, Microsoft Server 2003 and Microsoft SQL Server 2005 were used as operating system and DBMS, respectively.

B. Experimental results and analyses

1) Query processing times and metadata sizes with an increasing number of data providers

In this experiment, we measured the query processing times and metadata sizes of the four methods while increasing the number of data providers from 1 million to 10 million. We set the 'selectivity' of data providers and the number of usage purposes to 50% and 6, respectively. Here, the selectivity of data providers is the average rate with which they allow their information to be accessed for a given usage purpose.

As shown in Fig. 6, while the volumes of metadata of the LDHD, PBDM, and PBDM+G models increase linearly with the number of data providers, their difference in size becomes larger as the number of data providers grows. That is, the metadata volumes of the PBDM and PBDM+G models are approximately 30% and 6.5%, respectively, that of the LDHD model. However, as shown in Fig. 7, their query processing times do not differ so conspicuously. That is, although the PBDM+G model maintains less metadata than the LDHD model, the PBDM+G model incurs extra overhead by performing an additional join during query processing.

2) Query processing times and metadata sizes with an increasing number of usage purposes

Fig. 8 shows that the metadata size of the PBDM and PBDM+G models increases linearly with the number of usage purposes, but the LDHD model maintains a constant volume of metadata. This is because in the LDHD model, each data user possesses only a single usage purpose, and therefore its meta-size is not affected by the number of usage purposes but by the number of data users. In contrast to the LDHD model, because the number of usage purposes in the PBDM and PBDM+G models directly affects the size of the PreferenceTable(Purpose) and GroupMappingTable, the number of usage purposes directly increases the volume of metadata. However, even with 10 usage purposes, the PBDM and PBDM+G models require just 50% and 10% of the metadata volume required by the LDHD model.

As shown in Fig. 9, the query processing time of the PBDM and PBDM+G models is kept constant even though the number of usage purposes increases. This occurs because 1) the number of usage purposes added to user queries by the query modification procedure of the PBDM model is not related to the size of the metadata used in query processing, and 2) although the size of the GroupMappingTable increases in accordance with the number of usage purposes, the query processing time of the PBDM+G model does not change much due to the use of the index built on the GroupMappingTable.

3) Query processing times under increasing values of selectivity

Queries without any modification for privacy preservation have a constant processing time, as shown in Fig. 10. In such queries, data providers are assumed to be willing to disclose their entire data to all data users irrespective of their usage purposes. Therefore, such queries are assumed to have a selectivity of 100%. However, increasing selectivity values in the LDHD, PBDM, and PBDM+G models elevates the amount of data to be disclosed to data users, as well as the time required for query processing. The experimental result reveals that the queries of the PBDM+G model run 1.9% to 21% faster than those of the LDHD model.

4) Metadata volumes under an increasing number of data users

The growth of data users does not significantly alter the metadata size of the PBDM and PBDM+G models, as shown in Fig. 11. This is because these two models collect and store the information of user preferences only for each usage purpose. However, the LDHD model gathers the information of user preferences for every pair of data users and usage purposes. Consequently, the metadata size of the LDHD model rapidly increases as the number of data users grows. Both the PBDM model and the PBDM+G model maintain the greatest volume of metadata with 50 data users, but even their maximum metadata volumes are about 12% and 2.6% that of the LDHD model, respectively.

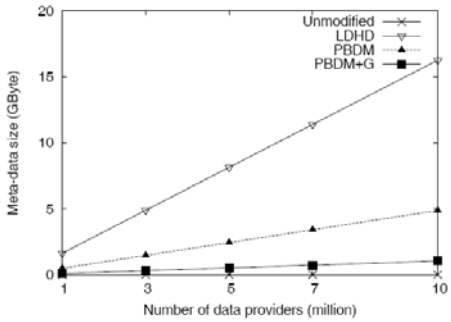


Figure 6. Metadata sizes of the four methods with an increasing number of data providers

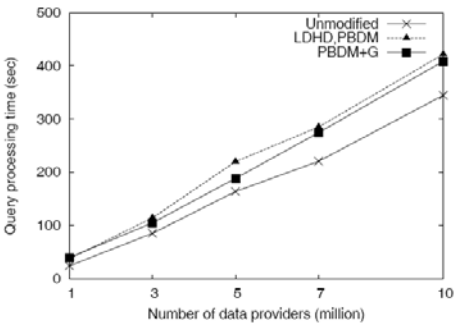


Figure 7. Query processing times of the four methods with an increasing number of data providers. We assume that the query processing time of the PBDM model is nearly identical to that of the LDHD model

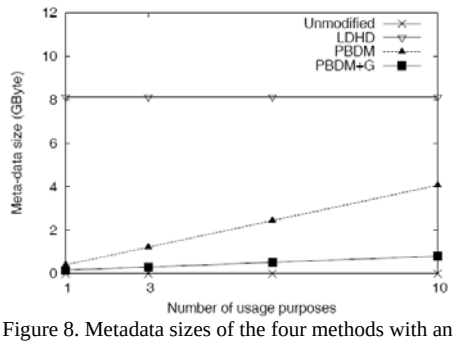


Figure 8. Metadata sizes of the four methods with an increasing number of usage purposes

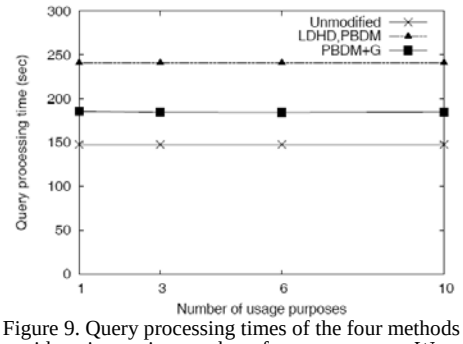


Figure 9. Query processing times of the four methods with an increasing number of usage purposes. We assume that the query processing time of the PBDM model nearly equals that of the LDHD model

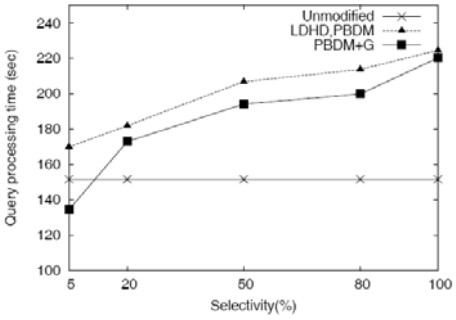


Figure 10. Query processing times of the four methods under increasing values of selectivity. We assume that the query processing time of the PBDM model nearly equals that of the LDHD model

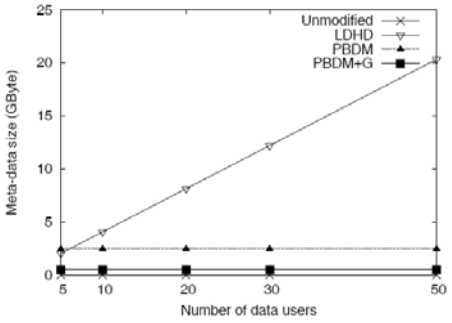


Figure 11. Metadata sizes of the four methods under an increasing number of data users

VII. CONCLUSION

In this paper, we first identified the requirements of database systems supporting privacy preservation, and then proposed an efficient and flexible database security model called PBDM+G. Based on the theoretical and experimental analyses, the proposed security model has been proved to be very practical and thus well suited to dynamic and/or large database environments.

The unique features that make our model very practical are summarized as follows:

(1) Rather than gathering metadata for every pair of data users and usage purposes, the proposed model collects metadata only for every usage purpose. As such, metadata need not be recollected when new data users are added to the system.

(2) The proposed model reduces the space required for metadata by employing grouping as sort of normalization approach. The grouping method simplifies the process of preference gathering by making data providers choose from a pre-defined set of preference groups.

(3) The proposed model enables efficient privacy checking by employing the 'left outer join' statement in the query modification algorithm.

Experiments reveals that the proposed PBDM+G model consumes at most 10% of the space needed for the LDHD model, a well-known database security model, while reducing query processing time up to 23.6%.

REFERENCES

[1] Office of the Information and Privacy Commissioner, Ontario, Data Mining: Staking a Claim on Your Privacy (1998).

[2] The Economist. The End of Privacy (May 1999).

[3] European Union. Directive on Privacy Protection (October 1998).

[4] Time. The Death of Privacy (August 1997).

[5] Online Americans More Concerned about Privacy than Health Care, Crime, and Taxes, New Survey Reveals, http://www.nclnet.org/pressessentials.htm

[6] R. S. Sandhu, E. J. Coyner, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models", *IEEE Computer* 29(2) (1996) 38-47.

[7] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Hippocratic Databases", *Proceedings of International Conference on Very Large Data Bases* (2002).

[8] K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu, and D. DeWitt, "Limiting Disclosure in Hippocratic Databases", *Proceedings of International Conference on Very Large Data Bases* (2004) 108-119.

[9] M. H. Harrison, W. L. Ruzzo, and J. D. Ullman. Protection in Operating Systems, *Communications of the ACM* 19(8) (1976) 461-471.

[10] C. J. McCollum, J. R. Messing, and L. Notargiacomo. "Beyond the Pale of MAC and DAC -Defining New Forms of Access Control", *Proceedings of IEEE Symposium on Security and Privacy* (1990) 190-200.

[11] P. P. Griffiths and B. W. Wade. An Authorization Mechanism for a Relational Database System, *ACM Transactions on Database Systems* 1(3) (1976) 242-255.

[12] S. Castano, M. G. Fugini, G. Martella, and P. Samarati, Database Security (Addison-Wesley, 1995).

[13] GA. K. Jones, R. J. Lipton, and L. Snyder. "A Linear Time Algorithm for Deciding Security", *Proceedings of IEEE Symposium on Foundations of Computer Science* (1976) 3341.

[14] C. Wood, R. C. Summers, and E. B. Feranadez, Authorization in Multilevel Database Models, *Information Systems* 4(2) (1979).

[15] D. E. Bell and L. J. La Padula, Secure Computer Systems: Mathematical Foundations and Model, *Technical report M74-244, MITRE Corp.* (1974).

[16] K. J. Biba, Integrity Considerations for Secure Computer Systems, *Technical report M76-372, MITRE Corp.* (1977).

[17] Osborn, S.L.: Role-based access control. In: Petkovic, M., Jonker, W. (eds.) Security, Privacy and Trust in Modern Data Management, (Springer, 2007), 55-70.

[18] Shipra Agrawal, Jayant R. Haritsa, B. Aditya Prakash, FRAPP: a framework for high-accuracy privacy-preserving mining, Data Mining and Knowledge Discovery, 18(1), (Springer, 2008) 101-139.

[19] D. DeWitt. The Wisconsin benchmark: Past, Present, and Future, The Benchmark Handbook (Morgan Kaufmann, 1993)