A System for Translating P3P Privacy Policies into FGAC Restrictions

Rakesh Agrawal† Paul Bird‡ Alvin Cheung†
Tyrone Grandison† Jerry Kiernan† Scott Logan‡ Walid Rjaibi‡

† IBM Almaden Research Center
650 Harry Road, San Jose, CA, USA
{ragrawal, alvin, tyroneg, jkiernan}@us.ibm.com

‡ IBM Toronto Lab
8200 Warden Ave.,Markham, ON, Canada
{pbird, silogan, wrjaibi}@ca.ibm.com

1. INTRODUCTION

The pervasive use of computing technology and the increased reliance on information systems have created a heightened awareness and concern about the storage and use of private information. In recent years a number of privacy-related guidelines and legislations have appeared worldwide, and compliance with these legislations has become an important corporate concern. Current methodologies used to address the disclosure compliance problem involve training individuals to be cognizant of the various regulations and changing organizational processes and procedures. However, these approaches are only a partial solution and need to be augmented with technology support.

In a paper that will be presented at ICDE 2005 [1], we proposed constructs for imbuing relational database systems with fine grained access control (FGAC) (see Figure 1). While earlier proposals address only database privacy or security [2, 4], these constructs have been designed to enforce both privacy and security in a consistent and unified manner.

In this demonstration, we present a realization of the FGAC functionality. We have implemented a translator that efficiently compiles privacy policies written in P3P [3], a W3C recommended high-level specification language, into FGAC restrictions for relational databases that restrict users' access to sensitive data. We will show how P3P privacy statements with different tag elements are handled by our translator to construct FGAC restrictions.

2. ARCHITECTURE

Figure 2 shows how the FGAC translator is incorporated into a relational database system. The translator accepts privacy policies, along with metadata stored in privacy catalogs, to generate FGAC cell restrictions. The restrictions are then integrated during query processing to modify input queries to conform to the disclosure constraints.


```
create restriction restriction-name
on table-x
for auth-name-1 [ except auth-name-2]
( (    (to columns column-name-list)
     | (to rows [ where search-condition ] )
     | (to cells (column-name-list [ where search-condition ] )+ )
  )
  [ for purpose purpose-list ]
  [ for recipient recipient-list ]
)+
command-restriction
```

Figure 1: Fine grained restriction syntax

The schema of the privacy metadata catalogs shown in Figure 2 used to drive the translation of P3P privacy policies into cell level restrictions is given below:

```
PR (    purp-recip char(18),
        p3ptype char(32),
        choice_tabname char(32),
        choice_colname char(32))

PT (p3ptype char (32), tabname char(32), colname char(32))
```

Table PR stores, for each purpose, the recipient and P3P data type pair, and the (table name, column name) pair that records individual user opt-in/out choice, should any choice be available for that combination. Table PT stores, for each P3P data type, the table names and column names that store values of these P3P types.

The syntax of an FGAC restriction is shown in Figure 1. It states that users in auth-name-1 except those in auth-name-2 are allowed only restricted access to table-x. The keywords **public** (i.e., all users), **group**, **role**, and **user** can be used to qualify the authorized names. Table-x can be any table expression.

A restriction can be specified at the level of a column , a row, or a cell. More than one restriction can be specified on a table for the same user.

A restriction may additionally specify purposes and / or recipients for which the access is allowed. If no purpose or

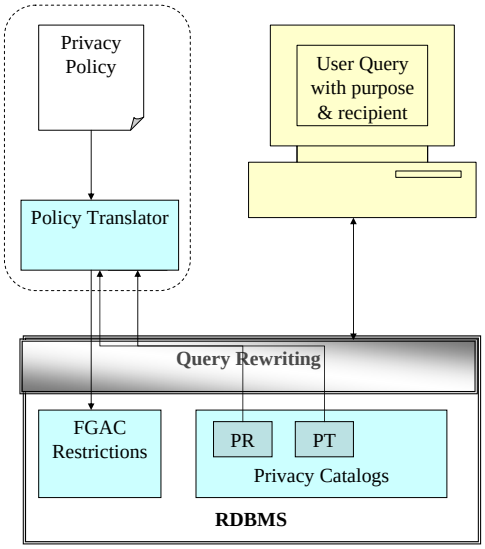

Figure 2: Implementation architecture (the dotted box represents our demonstration system)

recipient is specified, then the restriction applies to all purposes and recipients respectively. If a purpose or recipient is specified, the user's access is limited to only the specified purpose-recipient combinations.

3. SYSTEM DEMONSTRATION

The translator has been implemented in Java, and privacy metadata is stored in a database using IBM DB2 UDB Version 8.2 on Windows XP. We have also implemented a GUI using the Java Swing package for viewing the original policy and the restrictions that are generated, and a screenshot is provided in Figure 5.

3.1 Translation Example

Figure 3 is an excerpt from a sample privacy policy, and Figure 4 is the compilation results of the sample policy from our translator. Prior to translation, in the privacy metadata, we have associated the P3P datatype '#personal' with the columns name, SSN, and address while the '#medical' datatype is mapped to the columns xray and lifestyle fields, all stored in the Patients table. Meanwhile, the datatype '#financial' is associated with the columns acctnumber and balance. Thus, for the first privacy statement, one FGAC restriction statement is constructed with the corresponding P3P datatype mappings. However, for the second privacy statement, because it involves both the '#personal' and '#financial' datatypes, which span across two different database tables, two separate FGAC restrictions are generated, one for each table with corresponding columns associated with each datatype.

In general, if the P3P datatypes in a privacy statement map to columns in n different tables, n FGAC restriction statements will be generated, one for each table and relevant P3P datatype pair. The case for choice tables is similar. If the choices that pertain to a P3P datatype in a

```
...
<!-- Statement1 -->
<STATEMENT>
 <CONSEQUENCE>
  Encodes that personal and medical information
  can be accessed for emergency purposes
  by ourselves.
 </CONSEQUENCE>
 <PURPOSE>
  <other-purpose>
    Emergency
  </other-purpose>
 </PURPOSE>
 <RECIPIENT><ours/></RECIPIENT>
 <RETENTION><stated-purpose/></RETENTION>
 <DATA-GROUP>
  <DATA ref = "#personal"/>
  <DATA ref = "#medical">
   <CATEGORIES>
    <health/>
   </CATEGORIES>
  </DATA>
 </DATA-GROUP>
</STATEMENT>

<!-- Statement2 -->
 <STATEMENT>
  <CONSEQUENCE>
   Encodes that we and drug companies
   with the same data usage policies
   can access personal and financial
   information for new_drug_research.
  </CONSEQUENCE>
  <PURPOSE><develop/></PURPOSE>
  <RECIPIENT>
   <ours/>
  </RECIPIENT>
  <RETENTION><stated-purpose/></RETENTION>
  <DATA-GROUP>
   <DATA ref = "#personal"/>
   <DATA ref = "#financial">
    <CATEGORIES>
     <health/>
    </CATEGORIES>
   </DATA>
  </DATA-GROUP>
 </STATEMENT>

<!-- Statement3 -->
 <STATEMENT>
  <CONSEQUENCE>
   Encodes that we and drug companies
   with the same data usage policies
   can access financial
   information for new_drug_research
   on an opt-out basis.
  </CONSEQUENCE>
  <PURPOSE><develop/></PURPOSE>
  <RECIPIENT>
   <ours required="opt-out"/>
  </RECIPIENT>
  <RETENTION><stated-purpose/></RETENTION>
  <DATA-GROUP>
   <DATA ref = "#financial">
    <CATEGORIES>
     <health/>
    </CATEGORIES>
   </DATA>
  </DATA-GROUP>
 </STATEMENT>
 ...
```

Figure 3: A sample privacy policy

```
create restriction FGACRestrictionForStatment_1_1
on patients
for users public
to cells name, ssn, address, xray, lifestyle
for purpose Emergency
for recipient ours
restricting access to select


create restriction FGACRestrictionForStatment_2_1
on patients
for users public
to cells name, ssn, address
for purpose develop
for recipient ours
restricting access to select

create restriction FGACRestrictionForStatment_2_2
on finance
for users public
to cells acctnumber, balance
for purpose develop
for recipient ours
restricting access to select


create restriction FGACRestrictionForStatment_3_1
on finance
for users public
to cells acctnumber, balance
where exists (select 1 from
SysCat.Choice_Financial_1 p
       where p.ID = finance.ID and p.C1 = 1)
where exists (select 1 from
SysCat.Choice_Financial_2 p
       where p.ID = finance.ID and p.C1 = 1)
for purpose develop
for recipient ours
restricting access to select
```

Figure 4: FGAC restrictions generated from policy in Figure 3

privacy statement are stored across n different choice tables or columns, then n separate **where** clauses are generated, one for each choice table-column pair, as illustrated in the third privacy statement in Figure 3, where the P3P datatype '#financial' with the recipient 'ours' is associated with two different opt-out choices that are stored in choice tables SysCat.Choice_Financial_1, and SysCat.Choice_Financial_2. As a result, two **where** clauses are generated in the corresponding restriction statement.

With the growing impact of privacy legislations on the daily operation of information systems, the integrated and consistent management of security and privacy policies will become a primary concern for institutions. Our demonstration will help bring this requirement to the attention of database researchers and provide a foundation to further explore the integrated management of security and privacy policies.

4. REFERENCES

[1] R. Agrawal, P. Bird, T. Grandison, J. Kiernan, S. Logan, and W. Rjaibi. Extending relational database systems to automatically enforce privacy policies. In *21st Int'l Conference on Data Engineering*, Toyko, Japan, April 2005.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *28th Int'l Conference on Very Large Databases*, Hong Kong, China, August 2002.

[3] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification.* W3C Recommendation, April 2002.

[4] W. Rjaibi and P. Bird. A multi-purpose implementation of mandatory access control in relational database management systems. In *30th Int'l Conference on Very Large Databases*, Toronto, Canada, 2004.

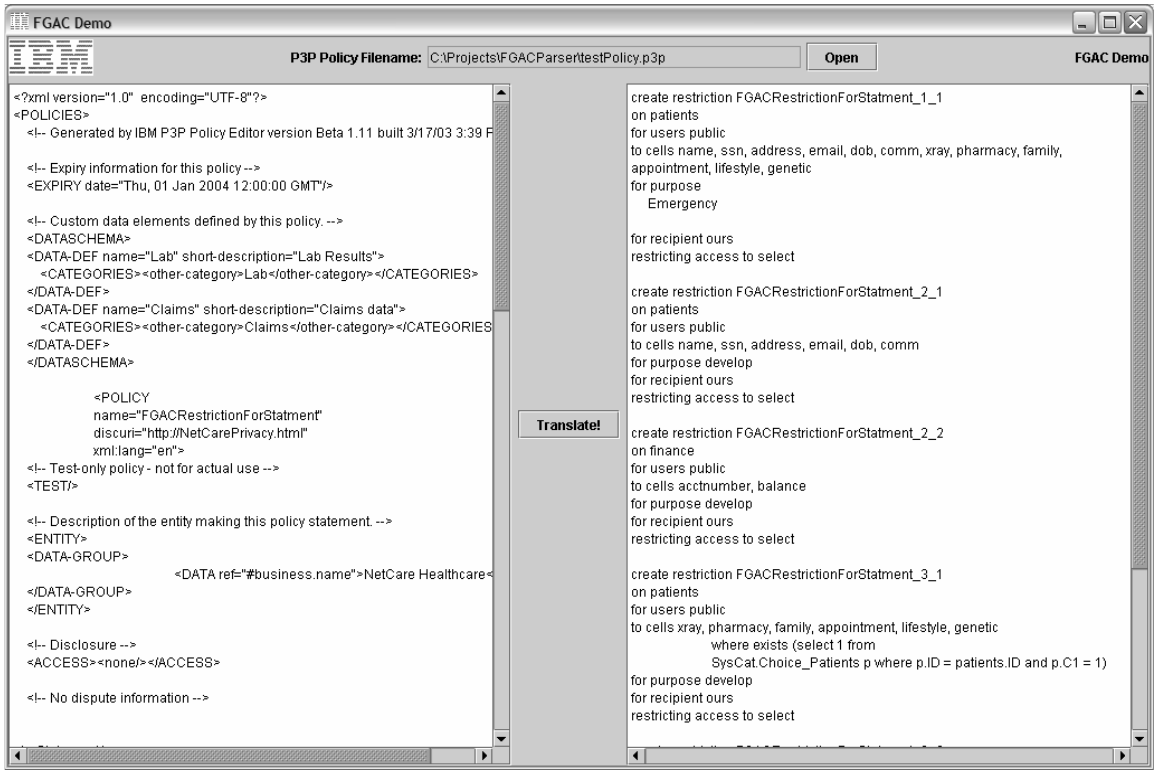

Figure 5: Demonstration system screenshot