A Three Layered Model to Implement Data Privacy Policies

Gerardo Canfora and Corrado Aaron Visaggio

Research Centre on Software Technology, University of Sannio
Viale Traiano 1, Benevento, Italy

Abstract. Many business services for private companies and citizens are increasingly accomplished trough the web and mobile devices. As such a scenario is characterized by high dynamism and untrustworthiness, existing technologies could be unsuccessful. This paper proposes an approach, inspired to the front-end trust filter paradigm, to manage data privacy in a very flexible way. Our approach has the potential to reduce the change impact due to the dynamism and to foster the reuse of strategies, and their implementations, across organizations.

1 Introduction

Nowadays, the number and complexity of processes which are accomplished throughout the web are increasing. Exemplar applications are e-government, e-procurement, and e-commerce, which are characterized by a continuous exchange of information: in such a scenario confidential data are more exposed to be collected lawlessly by humans, devices or software. Furthermore, the actors involved in these scenarios are often autonomous systems with a highly degree of dynamism [15]; negotiations are performed among multiple actors, and cross the boundaries of a single organization [10]. As a consequence, privacy of personal and confidential data is exposed to several threats [13]:

- data once collected will be persistent, also due to the decreasing cost of data storage;
- different devices record each event simultaneously from different viewpoints;
- the interpretation of the logged raw data for various purposes and the extraction of single events make the assignment of a valid privacy difficult;
- data is increasingly being collected without any indication about the "when" and the "how" [6].

The lack of a proper technology to protect unlawful access to data has been the root of costly damages. In 2001 82,000 cases of identity theft were reported, increased up to 126,000 in 2002: the increment is about 80% each year. As discussed in the section of the related work, the existing solutions show some limitations when applied in contexts characterized by high dynamism and a few opportunities to control data exchange: they are scarcely scalable, they cannot be used in untrustworthy transactions, or they propose too invasive data access mechanisms, which hinder flexibility. The realization and the adaptation of a data privacy policy is a process of transformation, which spans form the definition of strategies to properly protect data

Canfora G. and Aaron Visaggio C. (2007).
A Three Layered Model to Implement Data Privacy Policies.
In *Proceedings of the 5th International Workshop on Security in Information Systems*, pages 155-165
DOI: 10.5220/0002418301550165

up to the design of a supporting technology which implements the established policies. Such process includes three main stages.

At a first stage, a data privacy policy is described in natural language in a document which contains the rules to be adopted when dealing with sensitive data or, better say, information, concerning a specific environment. For instance, it is the case of laws that rule how to handle health data of patients, or banks' contracts that define how financial data should be treated. At a second stage, the general policy must be refined in specific strategies, in order to understand which kinds of actions could be performed on certain categories of data by some categories of users, and under which conditions. For instance, a strategy could prescribe that the information about infective illnesses of patients has high level of confidentiality. Finally, the established strategies need to be implemented with a suitable technology ensuring that accesses to the data repository are accomplished accordingly with the strategy.

This three-stages process can be very complex, depending on the data privacy policy and the static and dynamic characteristics of the accesses and of the data repository. It could become particularly critical and costly especially in the highly dynamic and untrustworthy contexts discussed above. This paper proposes a three-layered approach, whose main purpose is to provide the data manager with the capabilities of:

- reusing single parts of the strategies or implementations across different organizations, data domains, or data privacy regulations; and
- decoupling the three layers, in order to reduce the change impact, when some parts of privacy regulations, strategies, or implementation require modifications.

The paper proceeds as follows: related works are analyzed in section 2; the solution is introduced in section 3; section 4 discusses validation issues of the proposed approach. Section 5 roughly describes the software developed for implementing the approach; section 6 shows a case study; and, finally, conclusions are drawn in section 7.

2 Related Work

Different technologies have been proposed to preserve data privacy. The W3C Consortium developed P3P [17]. It synthesizes the purposes, treatment modes and retention period for data, but it does not guarantee that data are used accordingly to the declared policies. Consequently, it may be used only in trusted environments.

Researchers of IBM proposed the model of Hippocratic database [1]: it supports the management of information sharing with third parties, relying on ten rules for exchanging data. This technique degrades performances, as purposes and user authorization must be checked at each transaction. Memory occupation is a further matter, as the metadata could grow up fast.

The fine grain access control (FGAC) [2], is a mechanism designed for a complete integration with the overall system infrastructure. This kind of solution could be used only when constraints on data are few. Further solutions, like EPAL [3] and the one proposed in [14], allow actors of a transaction to exchange services and information

within a trusted context. The trust is verified throughout the exchange of credentials or the verifications of permissions to perform a certain action.

Anonymization techniques [5] [16] let organizations to retain sensible information, by changing values of specific table's fields. These techniques affect seriously data quality and may leave the released data set in vulnerable states. Further mechanisms of data randomization and perturbation [9] hinder the retrieval of information at individual level, and however are invasive both for data and applications. Cryptography is the most widespread technique for securing data exchange [8], even if it shows some limitations: high costs for governing distribution of keys, and low performances in complex and multi-users transactions.

3 The Proposed Approach

Our approach aims at translating a privacy regulation in a front-end trusted filter [12] which allows the access to data only when data privacy policy is not violated.

We make the assumption that a privacy policy can be expressed at least at three different levels of detail, or ***layers***.

The highest layer is represented by the **privacy regulation (PR)**, which is a normative document written in a natural language, that usually defines rules or laws concerning how sensitive data should be handled or delivered. Human beings are the intended target of this document.

The intermediate level is the set of **privacy objectives (PO)**, which are semi-structured statements describing how data could be used by users. Privacy objectives depends on domain semantics. For instance, a privacy objective could be: "enterprises can read curriculum data of students if curriculum is public".

The **data model** describes which are the entities involved within the data domain, and which are the relationships among them. The main purpose of data model is to show how the information chunks are related among each others, and how the elementary data contribute to make up the high level information. This linking is very important, as it permits to map information of high abstraction's level with explicit fields in the database, throughout the definition of elementary data.

A PO gets the following form:

<user><can |can not> <action> <resource> <condition>

Where:

<user>	Represents a specific category of data users, which could be a human, an application, or another system, which sends a query within a specific session
<can | can not>	Defines if the user has (or has not) the permission access to a resource
<action>	represents the kind of actions the application requires to perform, and could consists of reading, updating, or deleting a resource
<resource>	is a kind of data, at any level of abstraction, defined in the data model, and that could be obtained by data to retrieve by the repository
<condition>	describes a particular property to be verified in order to perform or deny the action

With regards to the previous example, the PO is:

<user>	"Enterprise"
<can | can not>	"can"
<action>	"Read"
<resource>	"curriculum data of students"
<condition>	"curriculum is public"

The entities used in the PO are semantically characterized, i.e. they should be defined within the data model and they should reflect existing entities of the data domain the application refers to. Consequently, a PO refers to specific kinds of users which have been previously categorized. The <action> tag refers to a set of operations which could be performed on the data. <resource> points out a particular information, which is not necessarily atomic, i.e. an elementary data. On the contrary, the assumption is that a resource is likely to be a complex data structure which aggregates different elementary data.

For instance, "*curriculum*" is a list of exams passed by a student. The entity "*exam*" is a record of elemetary data: "*Exam_Name*", "*Exam_Date*", "*Exam_Grade*", "*Name_of_Professor*".

The <condition> tag could be verified in different ways: (i) it could be contained in the *where* clause of the query; (ii) when the condition refers to a specific aspect of the database state, it could be necessary to launch a proper routine; and, finally, (iii) the condition could refer to a specific static or dynamic characteristic of users.

The lowest layer of the model is represented by the **privacy rules set** (or just "rules set" in the reminder of the paper) which implements a given privacy objective. A rule assumes the form of a query that the user can or can not send to the database. The rules are dependent on the specific database, unlikely the PO, which depends on the domain.

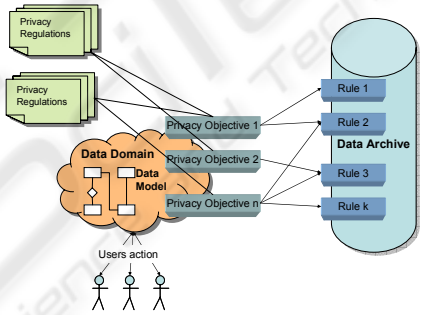

Fig. 1. The model applied.

A privacy rule follows this format:

No action [$Table_1.Field_{11},\ldots,Table_m.Field_{mn}$]
from $table_1,\ldots,table_m$
where {null | {Expression}}

As it is not the focus of this paper, the characterization of users will be not dealt with.

The following relationships could be established among the entities:

- A privacy regulation includes more than one PO, and some privacy regulations could share a subset of POs. This happens because the POs are related to the data domain.
- A PO includes more than one rule, and some objectives could share a subset of the rules set. This happens because the rules refer to a specific database.

Accordingly, both privacy objectives and the rules should be kept in dedicate collections, in order to facilitate reuse of both, when varying privacy regulations, but not the data domain, or when varying the privacy objectives, but not the database.

Figure 1 summarizes entities and relationships as they are used within the proposed model: privacy regulations contain rules and laws which regulate how users might access to data; privacy objectives are specific of a given data domain, and some privacy regulations can share some privacy objectives.

This eases the reuse of some POs for different privacy regulations, when they insist on similar data domains or involve similar entities. For instance, the PO: "Never deliver information on individual address", could be used in different privacy regulations. Reusing a PO could mean reusing a subset of the rules which implement it.

Each PO can be realized with one or more rules, which act at database level: rules are specific of a given database design. This entails that, if the same information could be obtained with a particular query, it is possible simply to add a rule which could ban that specific interrogation.

Thanks to the layered structure, adding, changing, or modifying a PO or a rule has a limited impact on the overall structure.

4 Validation of the Data Privacy Policy

The proposed approach needs to be validated properly, in order to be actually applicable. Validation phases are described below.

Completeness of Rules Set with Respect to the Database Design. The privacy policy is defined throughout a set of POs, which are implemented by a set of rules. As a PO indicates the access modes for handling a specific piece of information, the set of rules that implement that privacy objective must refer to all the database fields that contribute to form the information. Please consider that this is not a trivial property to verify, as the same information could be formed by fields belonging to different tables.

Definition 1. A Rule Set RS is PO-complete with respect to a PO if all the elementary data items that make up the information managed by the PO are completely covered by the RS.

If the RS is not complete, a malicious attack could exploit the fields not controlled by the RS to obtain information which could not be delivered.

Example. Let's consider this PO: "Enterprise can not read the final grade of students, if final grade is less than 100". Let's suppose that the correspondent RL include only the following rule:

- No select Name, FinalGrade From Students where FinalGrade<100.

If another relation exists where it is possible to obtain the same information with a query like this: “select avg(examgrade) from ExamLog where avg(examgrade) < 100 group by IDStudent”, then the RS is not complete and a new rule must be added:

- “No select avg(examgrade) From ExamLog group by IDStudent”.

Completeness and Correctness of the POs with Respect to Data Domain. The set of POs refers to all the information contained within a specific data domain. In order to make effective the privacy policy, the POs should be defined upon a complete and correct representation of the data domain. As a matter of fact, if some entities or some relationships are missing in the data model, or they are not correctly represented, the privacy policy could be less effective.

Definition 2. A set of POs is Domain-complete, if it includes all the entities and relationships which constitute the information concerned by the privacy policy.

Definition 3. A set of POs is Domain-correct, if the entities which constitute the information concerned by the privacy policy are correctly related among each others.

If a set of POs is not Domain-complete, neither Domain-correct, the implementation of the privacy policy could be unfeasible.

Example. Let’s consider the PO: “Enterprise can not read curriculum of students if curriculum is not public”. According to the data model, Curriculum is composed of the following data: exam, date, and grade. Let’s suppose that within the data domain the curriculum should include also information about work experience: in this case the PO is not complete.

Consistency of the Rules Set with the Existing Database. The set of rules should be compliant with the design of the database. They must refer to tables, fields, and relations which actually exist.

Definition 4. The RS is Database-consistent if all the rules refers to an existing database schema.

If the set of rules is not Database-consistent, the privacy policy could not be implemented.

Example. Let’s consider the following rule: “No select Name from table Students”. If the table Students exists, but it has not any “Name” field, than the RS is not Database-compliant.

5 The System

In order to apply the approach, a prototypal tool has been developed in the laboratories managed by the authors of the paper.

The tool supports the creation and the collection of the POs List for any Data Privacy Project. The tool helps to assess the syntactic correctness of POs; it provides a taxonomy to organize POs lists according to different data domains.

For each PO, and once that the data source has been selected, the tool automatically generates a list of rules, in order to support the user to define PO-complete RS. The user can also define rules on her own, as well.

The tool has algorithms for optimizing the RS: they reduce the RS size by aggregating rules or removing redundant ones. Routines for verifying that the RS is database-consistent are in place.

A monitoring engine quantitatively evaluates the impact of privacy policy on a database; it computes the following indicators:

- overall number of rules;
- overall number of rules per PO;
- overall number of rules per relation;
- average number of rules per relation;
- number of relations without any rule;
- number of rules automatically produced by the tool;
- percentage of rules accepted by the user.

6 Case Study

In order to obtain a preliminary validation of the proposed approach, we have applied it to a case study with the aim of evaluating the efficacy of the system. As this is only a preliminary analysis, we focused on some facets of the system's efficacy.

More specifically we aim at evaluating the capability of the system, while supporting the creation of a data privacy policy by:

- creating a set of rules which are compliant with the correspondent PO (briefly, ***Compliance,*** in the rest of the paper);
- determining a good level of tables' coverage by the automatic production of rules (briefly ***Database Coverage*** in the rest of the paper);

In the case study three databases were used:

- The Student Secretariat, in the administrative domain - 14 tables, 516 records;
- The FBI, in the criminological domain - 29 tables, 317 records;
- The Hospital in the medical domain - 14 tables, 413 records.

All the three databases are replications of existing repositories, realized by the students during laboratory sessions of different courses. For precision's sake:

- The Student Secretariat reproduces the system in place at our University to register information about students, lessons, courses, and exams (www.unisannio.it);
- The FBI was inspired to the application of the FBI official site for retrieving information about crimes, investigations, and wanted (www.fbi.gov);
- The Hospital replicates the informative system of a local hospital.

For each database, fifteen POs were defined, accordingly to the actual use scenarios of the three systems and the privacy requirements. These requirements were defined basing on the Italian law about personal data treatment (http://www.legge196.net/196.asp). This process was realized by the students who developed the system and supervised by the authors of this paper.

The "Compliance" goal was evaluated as the difference between the overall number of rules produced by the system and the size of the susbset of rules accepted.

$$Diff = Rules\_prod - Rules\_con,$$

where:

Rules_prod is the number of produced rules
Rules_con is the number of consistent rules

The descriptive statistics of Diff indicator is reported in the table 1.

Table 1. Descriptive statistics of the index "Diff".

Statistical indicator	FBI	Secretariat	Hospital
Max	7.000	7.000	5.000
Min	0.000	0.000	0.000
Mode	0.000	0.000	0.000
Average	0.733	0.800	0.600
CV	2.496	2.371	2.253
Devstandard	1.830	1.897	1.352
Curtosis	11.367	8.996	8.723
Median	0.000	0.000	0.000

For precision's sake, CV is an index of the sample dispersion, while curtosis evaluates the influence of tails on the sample distribution. As the CV suggests, the variability in the three samples is pretty close and low. It might be a good index about the quality of the rules produced by the system, since it could mean that the index is not far from the zero in the entire sample. The max values are high, and this points out that in some cases the tool generated a lot of rules which were not compliant. It is the case of POs which are stated at a very high level of abstraction, and consequently the system produced very generic rules which did not fit well with the intended purpose.

Let's consider the PO corresponding to the max value, shared by the three databases: "Not deliver information about the name of the person". The problem was the word "information": this term was too vague and the domain modelling was not enough accurate to describe it; it should be improved by increasing the accuracy of data model, or reducing the abstraction of the PO. The minimum value is equal to the mode and to zero in all the cases. This suggests that, with regard to the set of POs the system is very effective, as it produced all compliant rules. We defined another indicator, in order to take into account both the percentage of not compliant rules and the overall number of produced rules. To consider only the difference could be misleading. As a matter of fact, it is more important to have a difference equals to zero on *m+n rules* rather than on *m rules*. Or it could be more significant to have a difference of one on 10 produced rules, rather than to have a difference equals to zero on 1 rule produced.

The new indicator was defined as:

SinInd= Rules_prod*sin((rules_con/rules_prod)*90)

The descriptive statistics of SinInd indicator is reported in table 2.

Table 2. Descriptive statistics of "SinInd".

Statistical indicator	FBI	Secretariat	Hospital
Max	8.457	10.005	5.196
Min	1.000	1.000	1.000
Mode	1.000	1.000	1.000
Average	2.408	2.528	1.663
CV	1.038	0.981	0.689
Devstandard	2.501	2.481	1.147
Curtosis	2.831	6.899	6.323
Median	1.000	1.414	1.000

With this indicator, the variability in the sample is smaller than with the previous one: this suggests that this indicator provides a better picture of the stability of the results. Once again minimum and mode values are the same. As a matter of fact most policy objectives produced only one rule, and it was PO-compliant.

In order to have a better idea of the system's efficacy, it helps to examine the case of the max value for the three databases:

- 7 consistent rules out of 9 produced for the FBI;
- 8 consistent rules out of 11 produced for the Secretariat;
- 4 consistent rules out of 6 produced for the Hospital.

Concerning Database Coverage, the observed values for the three databases are reported in table 3.

The FBI database could be scarcely covered by rules, as only one third of tables (11 out of 29) are interested by rules of privacy. As a matter of fact, each table has in average one rule. This suggests that the policy of data privacy could need an enforcement.

Table 3. Metrics collected for the three databases.

Database	FBI	Student Secretariat	Hospital
Number of Rules	31	37	29
Number of Tables	29	24	14
Average Rules per Table	1.069	1.542	1.429
Number of Tables with no rules	11	9	5
Number of Privacy Objectives	15	15	15
Number of Tables without Privacy Objective	6	6	2
Number of Proposed Rules	44	45	26
Number of accepted Rules	31	37	20
Percentage of Accepted Rules	70%	82%	76%

For the Secretariat Database, the situation is similar: one third of tables is not covered by privacy rules. A major concern regards the only table with ten rules. This table should be properly analyzed by the data manager, and if it is the case, it could need a reengineering, in order to make it more usable. In the case of the Hospital database, the distribution of rules per tables is better, while the coverage is similar to the previous ones (5 out of 14).

7 Conclusions

The increasing migration to the web of transactions and services made urgent to have in place effective technologies for data privacy management. As the emerging scenarios are characterized by high dynamism and untrustworthiness, it is necessary that such technologies allow scalability, be not invasive, and foster evolution of technology and architecture.

This paper proposes a solution to be applied in highly dynamic, untrustworthy and scalable contexts, which implement the paradigm of *front end trust filter*. The data privacy policy is considered as a three-layered process consisting in the statement of the policy, the strategies for realizing such policy and the implementation, which applies the strategy at the level of applications and database.

This three-layered structure confers a high degree of flexibility which permits: (i) the reuse of strategies or implementations cross organizations and cross policies; and (ii) the reduction of the change impact due to modifications to database, technology, strategy, and regulations.

A case study was carried out in order to obtain preliminary validation of the system. The outcomes confirm the usefulness of the system in supporting the data privacy policy definition and maintenance. Two preliminary lessons emerged from the case studies. As first, privacy objectives should not be too generic, otherwise the automatic generation of rules could fail. As second, the set of privacy objectives derived from a regulation often needs to be enriched with additional ones, implicitly assumed by the regulation itself, or the automatic generation will leave a part of the database uncovered by mechanisms for preserving privacy.

Future directions include: (i) a larger investigation which focuses on further aspects of the system's effectiveness; and (ii) features for data domain modelling tailored on the processes of data privacy preservation.

References

1. Agrawal R., Kiernan,J., Srikant R., and Xu Y., 2002, Hippocratic databases. In *VLDB*, the 28th Int'l Conference on Very Large Database.
2. Agrawal R., Bird P., Grandison T., Kiernan J., Logan S., Rjaibt W., 2005 Extending Relational Database Systems to Automatically Enforce Privacy Policies. In *ICDE'05 Int'l Conference on Data Engineering*, IEEE Computer Society.
3. Ashley P., Hada S., Karjoth G., Powers C., Schunter M., 2003. Enterprise Privacy Authorization Language (EPAL 1.1). *IBM Reserach Repor*t. (available at: http://www.zurich.ibm.com/security/enterprice-privacy/epal – last access on 19.02.07).
4. Bayardo R.J., and Srikant R., 2003. Technology Solutions for Protecting Privacy. In *Computer*. IEEE Computer Society.
5. Fung C.M:, Wang K., and Yu S.P., 2005. Top-Down Specialization for information and Privacy Preservation. In *ICDE'05, 21st International Conference on Data Engineering*. IEEE Computer Society.
6. Langheinrich M.,2005. Personal privacy in ubiquitous computing –Tools and System Support. *PhD. Dissertation, ETH Zurich*.

7. Machanavajjhala A., Gehrke J., and Kifer D., 2006. l-Diversity: Privacy Beyond k-Anonymity. In *ICDE'06 22nd Int'l Conference on Data Engineering* . IEEE Computer Society.
8. Maurer U., 2004. The role of Cryptography in Database Security. In *SIGMOD, int'l conference on Management of Data*. ACM.
9. Muralidhar, K., Parsa, R., and Sarathy R. 1999. A General Additive Data Perturbation Method for Database Security. In *Management Science, Vol. 45, No. 10*.
10. Northrop L., 2006. Ultra-Large-Scale System. The software Challenge of the Future. *SEI Carnegie Mellon University Report* (available at http://www.sei.cmu.edu/uls/ – last access on 19.02.07).
11. Oberholzer H.J.G., and Olivier M.S., 2005, Privacy Contracts as an Extension of Privacy Policy. In *ICDE'05*, *21st Int'l Conference on Data Engineering*. IEEE Computer Society.
12. Pfleeger C.R., and Pfleeger S.L., 2002. *Security in Computing*. Prentice Hall.
13. Sackman S., Struker J., and Accorsi R., 2006. Personalization in Privacy-Aware Highly dynamic Systems. *Communications of the ACM, Vol. 49 No.9*.ACM.
14. Squicciarini A., Bertino E., Ferrari E., Ray I., 2006 Achieving Privacy in Trust Negotiations with an Ontology-Based Approach. In *IEEE Transactions on Dependable and Secure Computing,* IEEE CS.
15. Subirana B., and Bain M., 2006. Legal Programming. In *Communications of the ACM, Vol. 49 No.9.* ACM.
16. Sweeney L., 2002. k-Anonymity: A model for Protecting Privacy. In *International Journal on Uncertainty, Fuzziness and Knowledge Based Systems, 10*.
17. Platform for Privacy Preferences (P3P) Project, W3C, http://www.w3.org/P3P/ (last access on January 2007).