An Adaptive Privacy Management System For Data Repositories

Marco Casassa Mont, Siani Pearson, Pete Bramhall
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2004-211
November 18, 2004*

E-mail: marco.casassa-mont@hp.com, siani.pearson@hp.com, pete.bramhall@hp.com

privacy, privacy enforcement, adaptive system, information flow, data views, data protection, access control, accountability

This paper addresses the problem of dealing with privacy management of confidential data stored by enterprises and other organisations. We describe an innovative solution based on an adaptive privacy management system. In this system (arbitrarily complex) data structures are retrieved from standard data repositories, in such a way that parts of these data are obfuscated and associated with privacy policies. Data structures containing confidential data are "first class" objects that can be sent to other parties. Entities that try to access their content can be different from those entities that retrieve these objects. In particular, a Privacy Management Service decides what is visible at a given time for each specific request for content. The visibility of (and access to) the obfuscated data is adaptive, depending on the requestor, the context and purpose. Hence multiple "views" on a data structure can be provided by our system. Our research and development is work in progress; the aim of this paper is to share and describe our current results.

* Internal Accession Date Only
Approved for External Publication

An Adaptive Privacy Management System For Data Repositories

Marco Casassa Mont, Siani Pearson, Pete Bramhall

Trusted Systems Laboratory
Hewlett-Packard Laboratories
Bristol, UK
marco.casassa-mont@hp.com, siani.pearson@hp.com, pete.bramhall@hp.com

Abstract

This paper addresses the problem of dealing with privacy management of confidential data stored by enterprises and other organisations. We describe an innovative solution based on an adaptive privacy management system. In this system (arbitrarily complex) data structures are retrieved from standard data repositories, in such a way that parts of these data are obfuscated and associated with privacy policies. Data structures containing confidential data are "first class" objects that can be sent to other parties. Entities that try to access their content can be different from those entities that retrieve these objects. In particular, a Privacy Management Service decides what is visible at a given time for each specific request for content. The visibility of (and access to) the obfuscated data is adaptive, depending on the requestor, the context and purpose. Hence multiple "views" on a data structure can be provided by our system. Our research and development is work in progress; the aim of this paper is to share and describe our current results.

1. Introduction

Enterprises store large amounts of confidential data about their employees, customers and partners. On the one hand, accessing and managing this data is fundamental for their business: confidential information is retrieved, analysed and exchanged between people (and applications) that have different roles within an organization (or across organizations) to enable the provision of services and transactions. On the other hand, data protection and privacy laws, including [1,2,3], dictate increasingly strict constraints about how these data have to be protected, accessed and managed. Failure to comply with such privacy laws can have serious consequences for the reputation and brand of organizations and have negative financial impacts. There is therefore a need to reveal sensitive data but this must be done in a way that is legally compliant.

Privacy management technology can help achieve such a balance: this paper describes HP Labs' approach to addressing the problem above by providing an adaptive privacy management system for data repositories. Our main objective within this work is to enable adaptive access to confidential information based on the satisfaction of privacy policies with a minimal impact on data repositories in terms of required technological changes. The latter is important in order to aid the practical deployment of the system.

A privacy model is introduced, based on: a Privacy Virtualisation Layer used by people and applications to mediate their interactions with data repositories as dictated by privacy policies, and one or more Privacy Management Services (i.e. trust services run by organizations or trusted third parties) dealing with the enforcement of privacy policies. The process of disclosing confidential data is adaptive to contextual information. Our research and development is work in progress. In this paper we describe the main concepts underpinning our work and current results.

2. Addressed Problem

The key problem addressed in this paper is the management of privacy for confidential data stored by enterprises and other organisations.

Privacy management is not just a matter of authentication and authorization. When dealing with confidential (personal) data - among other things - it is necessary to capture the purpose of data, convey the consensus of the data owners (subjects) and make decisions on access requests based on the requestors' intentions.

Privacy policies can dictate additional terms and conditions under which access to confidential data can be granted: this involves the satisfaction of constraints and obligations which might require the processing of credentials, trust verification and management of contextual information.

In large organisations, people have different roles and skills: business tasks are achieved thanks to collaboration among these people. The rigid enforcement of privacy policies might create disruptions in business practices and introduce unacceptable burdens. For example, confidential data can be stored in a variety of data repositories. Only technical specialists might have the right skills to retrieve these data in a way that is meaningful for business people, marketing departments or strategists. Unfortunately, privacy policy constraints might dictate that these technical people must not access confidential data: in this case they would not be able to provide a service to the business people. Similar observations apply for applications and services run by different organizations within an enterprise.

Mechanisms are required to address both privacy requirements and business needs. Entities or applications must be enabled to retrieve confidential data by searching data repositories. The process of accessing confidential information has to be flexible and adaptive to contextual information and privacy policies.

An entity should not be prevented from acting on behalf of other people when searching and retrieving data, even if they cover different roles and have different privacy clearances. In such a case different views of this data must be provided, according to predefined privacy policies. In case of non-compliance to specific privacy policies, parts of this data might be removed or simply obfuscated.

3. Related Work

Relevant work in the area of privacy management for data repositories has been carried out in the area of data encryption. Mechanisms and solutions have been built to encrypt confidential data when it is stored in data repositories. Significant work in this space has been done with Translucent Databases [4]. Most of these solutions focus on the "confidentiality" and access control aspects: they have little flexibility in providing policy-driven mechanisms encompassing aspects beyond authentication and authorization i.e. dealing with data purpose, matching the requestors' intentions against this purpose, enforcing obligations, etc.

Seminal work has been done by IBM with their research on Hippocratic Databases [5], i.e. databases that include mechanisms for preserving the privacy of the data they manage. Their proposed architecture is based on the concept of associating privacy metadata (i.e. privacy policies) to data stored in data repositories, along with mechanisms to enforce privacy. The drawback of this approach is that it might require substantial changes to current data repository architectures, an approach that might take a long time and require substantial investment (of all the involved parties) to succeed. These changes include adding privacy metadata via additional database tables and using modified Java Database Connectivity (JDBC) data adaptors [5] that deal with these privacy metadata and interact with external privacy engines: this will require customers to buy upgraded versions of databases. In addition, this approach does not take into account that the management of privacy spans across the database boundaries: such management has to be carried out within a broader context as it encompasses aspects such as the management of enterprise-wide privacy policies, obligations and application/service-based privacy policies.

In terms of commercial products, the state of the art in this space is IBM Tivoli Privacy Manager [6,7]. This provides mechanisms for defining fine-grained privacy policies and associating them to data. Privacy policies are based on P3P [8] but they will evolve towards privacy authorization-based policies - based on the EPAL [9] specification, i.e. policies containing authorization constraints along with constraints on contextual information and intents. This approach addresses the privacy management problem purely from an access control perspective. It does not include additional aspects relevant for privacy management such as trust management and dealing with ongoing privacy obligations dictated by legislation and enterprise's guidelines.

Our approach differs from the above solutions in that it aims at leveraging current data repository technologies and reducing to the minimum the impact on them, in terms of required changes. In our approach, interactions with data repositories can still happen as usual but with the additional guarantee that confidential data is now protected and contextually released, in a fine-grained way, based on the fulfillment of associated privacy policies.

Additional relevant work has been carried out for privacy management in the area of data mining and statistical databases. In this context, the main goal is to prevent privacy violations when using data mining learning algorithms, data correlations and linking techniques. Current privacy management techniques involve the provision of statistical approaches (i.e. information is not returned as it is but it is statistically modified, for example to reflect average values), data obfuscation and knowledge hiding.

Our approach is complementary to these techniques. We focus on privacy management for traditional data repositories rather than techniques for On-Line Analytical Processing (OLAP) systems and data mining. Our main objective is to ensure that stored data is accessed in a privacy compliant way, in dynamic environments. Some of the technical approaches we introduce in this paper could also be applied in the context of data mining.

4. Our Solution

This section introduces the model underpinning our privacy management solution, discusses a few relevant scenarios and briefly describes technical approaches for its implementation.

4.1 Model

The model underpinning our solution consists of three basic components, as shown in figure 1:

- A Privacy Virtualisation System;
- A Privacy Management Service;
- Data structures containing confidential data along with associated privacy policies.

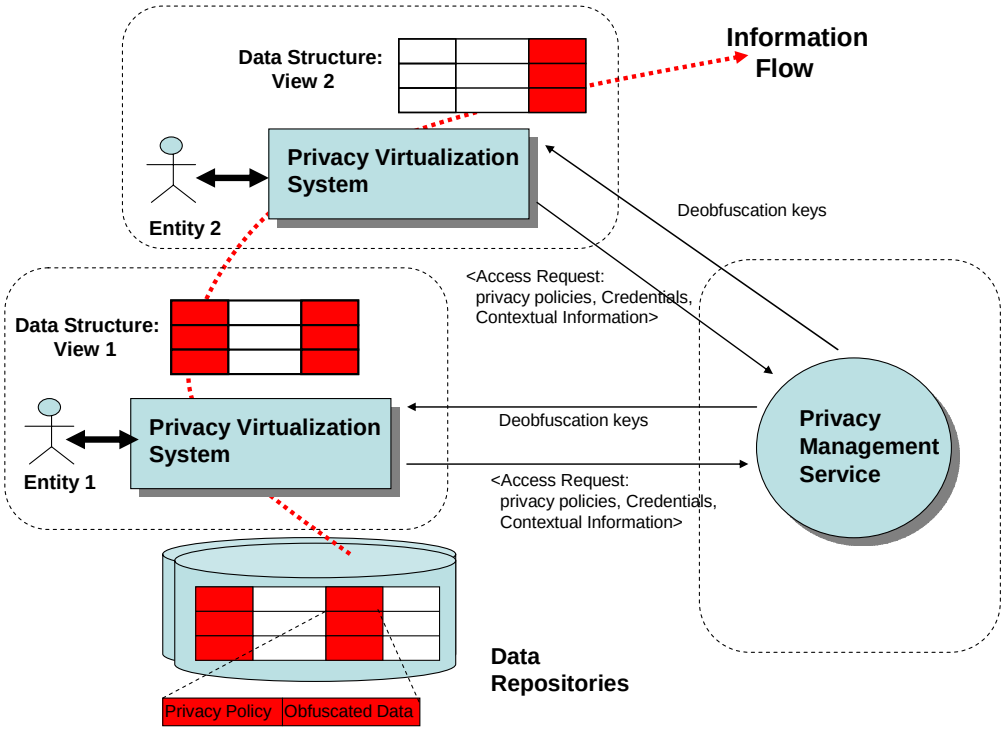

Figure 1: Model.

The Privacy Virtualisation System mediates the interactions between an entity (data requestor), data repositories and the Privacy Management Service. It allows users to retrieve confidential data from standard data repositories: part of these data can be stored in an obfuscated way and along with the associated privacy policies. Even if this might require some changes in the logical definition of data structures (i.e. different types of fields in tables, different LDAP classes' definitions, etc.) in order to store encrypted data and the associated privacy policies, no technological changes are required for data repositories.

Once retrieved, confidential data can be represented via data structures (i.e. "first class" objects) that simplify their portability in case of their transmission. Such a data structure contains the retrieved confidential information along with the associated privacy policies. It can be transmitted to other parties: the entity that accesses these data can be different from the entity that retrieved it.

The Privacy Management Service decides which confidential information can be accessed by an entity at a specific point in time by taking into account the specific request, relevant privacy policies and requestor's credentials. When access is granted, the Privacy Management Service discloses decryption keys to the requestor to enable the deobfuscation of confidential data.

The process of accessing obfuscated data is adaptive, depending on the requestor's credentials, relevant privacy policies, current context and data purpose. Figure 1 shows an example where different "views" of confidential data are provided by our system to different requestors. Confidential data can be retrieved by people and applications that have no rights to access its content (i.e. they do not satisfy privacy policies) but are in charge of querying data repositories on behalf of other people: in this case the content is not de-obfuscated. Nevertheless, the obfuscated data can be sent to other entities that can access their content if they satisfy the associated privacy

policies.

This basic model can be extended and adapted to a variety of scenarios including enterprise and inter-enterprises contexts. In particular the Privacy Management Service can be provided by an organisation for internal consumption or by one or more external trusted third parties, to enable multi-party interactions and at the same time increase the overall trust and accountability. For more details about the role of trusted third parties in such systems see [15].

4.2 Scenarios

This sections briefly describe a few scenarios where our solution adds value in the management of privacy for confidential data:

- **Enterprise Scenario**: an enterprise collects confidential data about employees, customers, partners, etc. People (or applications/services), with different roles and objectives might need to access this confidential information. Roles played by people include IT technicians, researchers, marketing people, project managers and HR people. The kind of confidential information they can access must depend on their role, their declared intent, purpose of the stored data, enterprise policies, legislation and specific customers' (opt-in and opt-out) policies;
- **Federated Identity Management Scenario**: Confidential information can be sent from a service provider *A* to a service provider *B* in the context of multi-party electronic interactions driven by a transaction. Depending on who initiated the transaction (customer, service provider, etc.), the purpose of data and also customers' policies, a subset only of the whole data may be accessed and sent to the other parties, as dictated by privacy policies. For example policy constraints could dictate that specific portions of confidential data cannot be sent outside an organisation for marketing reasons or that it can only be sent to a predefined set of organisations to enable customers' transactions.
- **Healthcare scenario**: it is important to have access control on a patient's medical record. Administrative staff, doctors, nurses, lab technicians, insurance providers, and researchers may have access to some but not necessarily all of a patient's information. Access to information depends on the purpose of data, the intention of the entity trying to access this data and the satisfaction of any specific fine-grained patient's preferences.

By using our approach we are able to associate fine-grained privacy policies to obfuscated confidential data and force requestors to be compliant to these policies if they want to access the data.This can be achieved in a flexible way, without *a priori* preventing the various entities from interacting, as dictated by business processes.

State of the art solutions can provide censored responses where private information is stripped out. Our solution can do this as well; the main competitive advantage of our approach (for example against Translucent Databases or IBM Hippocratic Databases) consists of the fact that data could be retrieved by people who might not be entitled to access confidential parts of it but are authorised to collect and organise this information on behalf of other people (who might have the right to access it). These data are obfuscated and their deobfuscation is subject to the fulfilment of privacy policies. As a consequence, incremental disclosure of confidential data can be obtained by requestors by providing the right credentials and satisfying privacy constraints.

Compared with traditional "views" on data (for example views on database tables), our approach reduces the need for defining a broad set views to accommodate multiple different cases, depending on requestors' capabilities and clearance: access and privacy constraints are direclty associated to data and dictate what can be seen at any point in time.

4.3 Technical Approach

Figure 2 describes the high-level architecture of a system implementing our model:

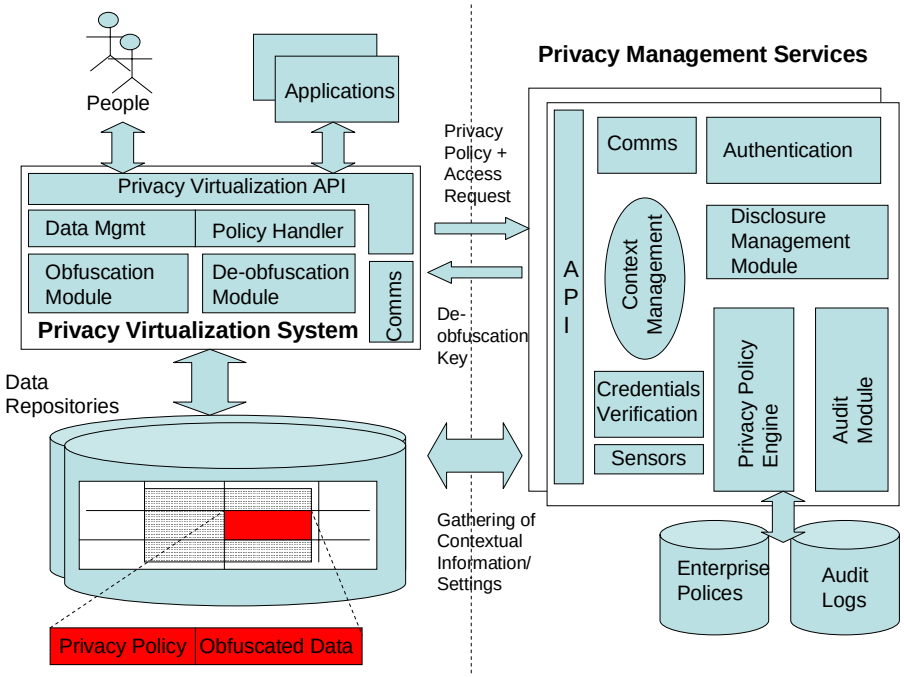

Figure 2: High-level Architecture.

Technical details related to privacy policies, storage of confidential data and architectural components follow.

4.3.1 Privacy Policies and Storage of Confidential Data

Our solution enables the storage of confidential data in data repositories by using an obfuscated format, along with fine-grained privacy policies (these policies dictate the terms and conditions under which the content can be deobfuscated and accessed). A basic form of privacy policy would consist of text describing a list of conditions and constraints (strings of characters). Examples of these policies can be found in [15]. Additional examples will be described later in this section.

The content of an obfuscated field (for example in a database record) can be represented as:

<privacy policy, Encryption(privacy policy data), Encryption(private data)>

This includes the encrypted (obfuscated) data, the relevant privacy policy and additional information to check for the integrity and stickiness of this policy to the data.

The specific format used to represent privacy policies is not a major issue and will not be described in this paper. However it is important to stress the fact that the format used to represent a policy must be flexible enough to express the following aspects:

- **privacy**: policies define conditions and constraints on how data must be handled, disclosed to other parties, protected, etc:
- **authorization**: policies dictate who can access what and under which conditions;
- **obligation**: policies define the constraints that need to be fulfilled potentially over a long period of time (in case of data retention or data deletion policies);
- **preferences**: policies define customers' preferences when multiple choices are available, for example in the way confidential information must be handled, disclosed or used;

- **trust**: policies dictate trust requirements to be satisfied by the involved parties;
- **control**: policies allow people to be involved in the management and monitoring of their data for example by explicitly asking to be notified each time their data is disclosed.

A few examples of policies follow. They reflect a user's perspective and they are related to a scenario where customers' data is stored and accessed by members' of an organization:

- Entities can access my data subordinate to the fact that their intent matches the "e-commerce transaction" purpose;
- Do not disclose any of my personal details to entities with identities X, Y, Z;
- Allow the access of these data only when dealing with entity W;
- Notify me via e-mail, every time you use some of my identity information;
- Ask for my authorization (via a predefined communication channel) every time you need to disclose this attribute to a third party;
- Interact with this trusted third party and state your intentions in order to obtain the current values of these attributes. You will be audited.

The above policies reflect customers' constraints and (for simplicity) they are expressed in natural language. Notice that constraints might require the fulfillment of actions involving the data owners, such as notifications or explicit requests for authorization. Different kind of policies can be used to express internal enterprise guidelines and privacy requirements.

In our system privacy policies are written in a formal language (via logical expressions and constraints) in a way that they can be programmatically interpreted.

It is important to ensure that a privacy policy sticks with the encrypted data and that this link cannot be broken. In our system the stickiness of policies to identity information is obtained by obfuscating the identity information in a way that its de-obfuscation is a "function of" the associated policies. Any tampering with these policies prevents the de-obfuscation of data. Different cryptographic techniques are available: they can use either traditional public-key cryptography or identifier-based encryption (IBE) cryptography. In both cases only the Privacy Management Service can issue the correspondent decryption key:

- In case public key cryptography [10] is used, the Privacy Management Service (PMS) publishes its (certified) public key (and, of course, it keeps the correspondent private key as a secret). A symmetric key is generated by the Privacy Virtualisation System (PVS) and used to encrypt the data. The symmetric key and a hash value of the associated policies are encrypted in a package [11] by using the PMS public key. The overall information (encrypted data, clear text policies and package) is stored. The PMS is the only entity that can decrypt the above package, check for the integrity of the associated policies, check for their compliance and eventually disclose the symmetric key.
- The alternative approach is based on IBE technology [12,13]. Any kind of string (including text, pictures, terms and conditions, etc.) can be used as an IBE encryption key. Privacy policies can be used for this purpose. The correspondent IBE decryption key can only be generated by the PMS as it is the only entity that has the "secret" necessary for doing it. The PMS will check the compliance of a requestor with these policies. The generation of IBE decryption keys can be postponed in time i.e. until they are actually necessary for decryption purposes. Any tampering with the IBE encryption key will make impossible for the PMS to generate the correct decryption key.

In terms of implementing the "stickiness of policies" to confidential information, the above mechanisms are conceptually equivalent.

4.3.2 Privacy Virtualisation System

The Privacy Virtualisation System provides APIs to people and applications to mediate the storage and retrieval

of confidential data. This API, at the very base, consists of an extension of traditional data repositories API (such as the API to access a relational database via SQL commands) which includes the possibility to pass or retrieve data along with privacy policies and the declared intention (i.e. the reason for making this request). However, this extension does not require changes to current data repositories.

For example, in case of access to a relational database, two basic interactions can happen:

- Storage and update of confidential data: in addition to traditional INSERT or UPDATE SQL commands, the privacy API allows users to specify the association of privacy policies to the data;
- Retrieval of confidential data: traditional SELECT queries can be asked to the database via the Privacy Virtualisation API. The Privacy Virtualisation System intercepts these queries and interacts with the Privacy Management Service to deobfuscate data. The actual deobfuscation will depend on the current context, user's credentials and privacy policies. The answer to the query could be provided either via a traditional database result set (where part of the data could be obfuscated) or via an explicit "data structure", based on the XML format.

The representation of query results via an explicit XML-based *"data structure"* allows a "transportable" representation of the result: this can include data in clear, obfuscated data and the associated privacy policies. This data structure can be transmitted to other parties and accessed via the mediation of the Privacy Virtualisation System (if the associated privacy policies are satisfied).

Eventually, applications and services need to be modified to be privacy-aware and to fully leverage this privacy API and handle privacy policies. This is particularly important for the storage of confidential data via the Privacy Virtualisation System, as only in this way data will be stored according to privacy criteria (e.g. data obfuscation). In absence of this, they can still be able to access data as usual (no changes are made to data repositories), except for confidential data.

In addition to the virtualisation API, the Privacy Virtualisation System consists of the following core components:

- **Data management module**: it is the component in charge of formatting data in a proper way, depending on the underlying data repository and the requested privacy policies. It provides a data "translation" service;
- **Policy handler module**: it is an interpreter of privacy policies. It is the component that interacts with the Privacy Management Service and ensures that the right information is provided to this service in order to obtain the de-obfuscation keys. It can be built in a way that the communication with the Privacy Management Service is optimized i.e. it happens only when it know it can satisfy the relevant privacy policies;
- **Obfuscation/Deobfuscation modules**: these modules are in charge of dealing with the encryption and decryption of confidential data, as described above;
- **Communication module**: this module enables secure communication with the Privacy Management Service.

4.3.3 Privacy Management Service

The Privacy Management Service is in charge of enforcing privacy policies associated to confidential data. As anticipated, confidential data can be retrieved by entities and applications in an obfuscated way. The access to the deobfuscated data is mediated by one or more Privacy Virtualisation System that interacts with the Privacy Management Service.

At the very base, the Privacy Management Service verifies that privacy policies are fulfilled before providing the keys for de-obfuscating confidential data. Any disclosure of keys is audited and monitored.

The Privacy Management Service consists of the following core components:

- **Communication module**: this module enables secure communication with the Privacy Virtualisation System and other parties;

- **Authentication module**: this module is in charge of authenticating requestors, in case their identities are important to enable the disclosure porcess, as dictated by privacy policies;
- **Credential verification service**: this module is in charge of verifying the integrity and validity of digital credentials, i.e. certified information (including identity and attibute credentials);
- **Context management module**: this module is in charge of storing contextual information, relative both to specific interactions and the general situation;
- **Sensors**: sensors can be used by the Privacy Management Service to gather additional up-to-date contextual information. For example, a sensor might deal with the gathering of trust measures from Trusted Computing Group (TCG) enabled platforms [14]: privacy policies might dictate that confidential data can only be accessed and manipulated by TCG platforms [14]. Add-ins can be deployed in the Privacy Management Service to extend the privacy enforcement mechanisms.
- **Disclosure management module**: fundamentally, this module is in charge of disclosing decryption keys. It interacts with the privacy policy engine to get the authorization to do this, once all the privacy policies are satisfied;
- **Privacy policy engine**: it is the privacy policy interpreter. The interpretation process drives its interaction with the Privacy Virtualisation System, sensors and the disclosure management module;
- **Audit**: this module logs all the interactions happening with requestors, in particular related to the disclosure of decryption keys. The audil log has, at least, to be tamper evident. Collecting auditing inforamtion is fundametnal to enforce accountability and ensure that, in case of privacy violations, forensic analysis can be done.

The content of obfuscated data can be incrementally deobfuscated, at different stages, by providing the Privacy Management Service with the required information (additional credentials, etc.).

The disclosure process is adaptive and driven both by privacy policies and contextual information. Contextual information can be very rich, including not only users' credentials and declared intents, but also system information, measures of trust of the requestors' platforms, historical information, etc. It is important to notice that the disclosure of confidential information can modify the current context and, as a consequence, enable/disable sets of privacy policies and influence future disclosures.

The Privacy Management Service can be deployed either remotely or locally to the site where the data repository is located. It could also be provided by a trusted third party to enable multi-party interactions and ensure a consistent enforcement of privacy policies.

In a more advanced scenario, privacy policies can ask the Privacy Virtualization System to interact with multiple Privacy Management Services (each of them having specific competences) in order to access obfuscated data.

5. An Example Illustrating our Approach

This section describes a simple example to illustrate our approach and related concepts. In this example, we consider a scenario where an airline maintains data about its customers in a customer table within its database. Each record of this table includes fields for:

- unique (internal) customer ID;
- customer name;
- customer address;
- customer credit card number;
- customer country;

- customer flight preferences;
- customer data usage preferences;
- customer gender.

The airline decides to implement a privacy policy that restricts viewing of the customer name field, customer address field and credit card number field. A selection of the requirements defined by the policy includes:

- all fields are to be viewable by members of the customer service department;
- the credit card number must be readable only by accredited personnel or systems within the account department;
- the name and address fields may be readable by the advertising department only if approved in the customer's data usage preferences.

During the implementation of our system, the schema of the customer table is not changed. Thus, any reports, SQL queries or the like that have been written for use with the database are not affected. The data in the data fields selected to be private is replaced with an obfuscated version of the data preceded by policy data defining the criteria that must be satisfied to view the respective data (in doing this data types might have been modified).

A basic form of privacy policy would consist of text describing a list of conditions and constraints (strings of characters). An example policy for the "address field" could be:

access granted if requestor.department = {customer_service }

OR if (requestor.department = {advertising} and data_usage = "Y")

The "policy data" preceding the obfuscated data may be the text of the policy itself, a link, such as a URL, to the policy text or an encoded version of the policy or some other value via which the private data mediating system can obtain the policy criteria (or at least details of data required for submission to the privacy manager to determine whether the policy is satisfied). The privacy manager or some other central entity may store the policy details.

Thus, when a member of the advertising department wishes to run a mail merge, he/she runs the SQL query:

select * from customer_table where customer_country = "uk"

All data fields would be returned for those entries having a customer country "uk" but the credit card field would be obfuscated (with no possibility of a member of the advertising department being able to de-obfuscate it) and the name and address fields would only be de-obfuscated if the respective data usage field permitted.

If the airline decided to restructure its database in the future (for example moving the credit card number field to a separate table indexed by unique customer ID), no change to the data privacy system would be needed.

Once retrieved, private data can be stored and/or represented via data structures that simplify their portability in case of their transmission. Such a data structure would contain any retrieved confidential information along with the associated privacy policies. One implementation of such a data structure would be in XML. A portion of XML data structure (including a header section and a record section representing an example record extracted as the result of the SQL query discussed above) is shown below.

```
<extracteddata>
  <privacymanager>125.18.219.66</privacymanager>
  <mediator>www.policysite.org/mediator.jar</mediator>
   <record>
        <customerID>123857841</customerID>
        <customername>Jane Doe</customername>
        <customeraddress>
               <street>123 Long Ave.</street>
               <city>New York</city>
               <state>NY</state>
               <zip>12345-0000</zip>
        </customeraddress>
        <customercreditcardnumber>www.policysite.org/12568.pol,MTM0VF
        9F5E$R96%K#$PCP3$QCP04T#2T</customercreditcardnumber>
        <customercountry>USA</customercountry>
        <customerflightpref>Window,Vegitarian</customerflightpref>
        <customerdatausage>Y</customerdatausage>
        <customersex>F</customersex>
   </record>
</extracteddata>
```

The field “privacymanager” defines the IP address to be used to contact the privacy manager responsible for controlling access to obfuscated data. The mediator field points to a location where a Java application that functions as a private data mediating system can be downloaded. The “customercreditcard” field is obfuscated in the manner described above and includes the obfuscated data preceded by a URL to the policy held on a web server.

The data structure can be transmitted to other parties such that the entity that accesses the private data can be different to, and possibly in an organization remote from, the entity that retrieved it.

The representation of data via an explicit XML-based data structure allows a “transportable” representation: this can include data in clear, obfuscated data and the associated privacy policies. This data structure can be transmitted to other parties where private data may only be made intelligible via the mediation of a private data mediating system (if the associated privacy policies are satisfied). The XML data structure may include a URL or other instructions detailing where a version of the private data mediating system can be obtained to address the eventuality that the receiving system does not include this functionality

6. Discussion

It is the case that our Privacy Virtualisation System can potentially be bypassed as requestors could try to access data by directly querying the data repositories or by accessing the content of files (if they have the basic access control rights). However, in this case, any obfuscated data is going to be unintelligible. This forces the requestor to interact with the Privacy Management Service as dictated by the associated privacy policies.

A more problematic issue arises because once confidential data have been disclosed to a legitimate requestor (that satisifed the associated privacy policies), it may not be possible to prevent this entity from misusing these data. At this stage also the association of sticky policies to data can be broken. Unfortunately, this is a common problem for systems that must enforce privacy and at the same time must release confidential data. With our approach we ensure that sticky privacy policies are strongly associated to data at least until the first disclosure happens. Afterwards our approach can mitigate the involved risks by auditing disclosures and the context where they happened. Audit logs increase the accountability of the involved entities and can be used for forensic analysis in case of detected privacy violations. In the future, it is likely that further controls will be available: most notably, if the requestor’s platform includes technologies such as security-enhanced operating systems (OS) and Trusted Computing Group (TCG) technology, these could potentially be used to control the use and propogation of

deobfuscated data, for example by OS-level checking over whether certain operations are allowed on specific (tagged) data and hardware-based control over the use of the data (such as only allowing it to be accessed within a trustworthy software state). More details on addressing the above issues can be found in [15].

An open question that needs to be addressed is the impact of our solution in terms of efficiency and its overall workability. The fact that confidential data is obfuscated/deobfuscated and a remote access to the privacy obligation service is required can introduce delays, although these operations only need be performed periodically and cryptographic and design choices can be made so as to minimise the time taken to perform operations that need to happen in real time.

We need to fully understand how applications and services will deal with the association of privacy policies to data. This is definitely work in progress.

Another important aspect that needs to be explored further is the overall lifecycle management of privacy policies associated to confidential data, including their renewal and modification. The management of keys is strictly related to the management of policies as decryption keys will be issued based on policy fulfillment. By changing a policy, our system can automatically change the associated encryption key. Revocation of keys and one-time usage of keys have to be addressed in this context. Related to these aspects, we are currently looking at ways to change encryption keys based on successful disclosures of data. This could be done via a combined interaction between the Privacy Management Service and the Privacy Virtualisation System where the Privacy Management Service asks the Privacy Virtualisation System to change the encryption keys at the disclosure time.

We are currently researching in this overall space and developing a prototype of our solution. Technology like traditional public key cryptography or identifier-based encryption (IBE) can be used to provide the required encryption and decryption mechanisms of confidential data and implement the concept of “sticky policies”. We can leverage TCG-enabled trusted platforms to provide further trust about contextual information. We have already developed policy engines able to release decryption keys based on the fulfillment of policies.

Aspects of our model might be further investigated and built in the context of the EU Framework VI, PRIME project - Privacy for Identity Management in Europe [16]. Our immediate objective is to research and implement a Privacy Virtualisation System that is as transparent as possible to applications and users and can leverage traditional relational databases and file systems.

7. Conclusions

This paper describes an innovative approach to deal with an adaptive mangement of privacy for confidential data. The disussed solutions, based on a Privacy Virtualisation layer and Privacy Management Services, allow an incremental disclosure of confidential data depending on the satisfaction of privacy policies, with minimal disruption to common business interactions. Confidential information can be retrieved and transmitted between people that potentially have the right to access only parts of it: different views (in the sense of visible data) of this information are provided, depending on the requestors’ credentials, the context and privacy policies. This is the main advantage of our approach if compared with current solutions.

Our reasearch and development is work in progress. Part of this research could be carried out within the context of the PRIME project, an international project on identity and privacy management funded by the the European Union.

8. References

[1] C. Laurant, 2003, Privacy International - Privacy and Human Rights 2003: an International Survey of Privacy Laws and Developments, Electronic Privacy Information Center (EPIC), Privacy International. http://www.privacyinternational.org/survey/phr2003/

[2] OECD, 2001, OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. http://www1.oecd.org/publications/e-book/9302011E.PDF

[3] Online Privacy Alliance, 2004, Guidelines for Online Privacy Policies. Online Privacy Alliance, http://www.privacyalliance.org/

[4] P. Wayner, 2002, Translucent Databases, Flyzone Press

[5] IBM, 2004, Hippocratic Databases, http://www.almaden.ibm.com/software/quest/Projects/hippodb/

[6] IBM, 2004, IBM Tivoli Privacy Manager, http://www-306.ibm.com/software/tivoli/products/privacy-mgr-e-bus/, 2004

[7] IBM, 2004, IBM Tivoli Privacy Manager, online technical documentation, http://publib.boulder.ibm.com/tividd/td/PrivacyManagerfore-business1.1.html

[8] W3C, 2004, P3P specification, http://www.w3.org.P3P/brochure.html

[9] IBM, 2004, The Enterprise Privacy Authorization Language (EPAL), EPAL 1.1 specification, http://www.zurich.ibm.com/security/enterprise-privacy/epal/

[10] R Housley, 1999, W. Ford, W. Polk, D. Solo, RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL profile, IETF

[11] RSA, 1997, PKCS#7, Cryptographic Message Syntax Standard, http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/

[12] D. Boneh, M. Franklin, 2001, Identity-based Encryption from the Weil Pairing, Crypto 2001

[13] C. Cocks, 2001, An Identity Based Encryption Scheme based on Quadratic Residues. Communications-Electronics Security Group (CESG), UK

[14] S. Pearson (ed.), 2002, Trusted Computing Platforms, Prentice Hall

[15] M. Casassa Mont, S. Pearson, P. Bramhall, 2003, Towards Accountable Management of Privacy and Identity Management, ESORICS 2003

[16] EU Framework VI PRIME Project, 2004, Privacy and Identity Management for Europe, http://www.prime-project.eu.org/