Provenance-based Auditing of Private Data Use

Rocío Aldeco-Pérez and Luc Moreau
School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
raap06r@ecs.soton.ac.uk, *l.moreau@ecs.soton.ac.uk*

Abstract

Across the world, organizations are required to comply with regulatory frameworks dictating how to manage personal information. Despite these, several cases of data leaks and exposition of private data to unauthorized recipients have been publicly and widely advertised. For authorities and system administrators to check compliance to regulations, auditing of private data processing becomes crucial in IT systems. Finding the origin of some data, determining how some data is being used, checking that the processing of some data is compatible with the purpose for which the data was captured are typical functionality that an auditing capability should support, but difficult to implement in a reusable manner. Such questions are so-called provenance questions, where provenance is defined as the process that led to some data being produced. The aim of this paper is to articulate how data provenance can be used as the underpinning approach of an auditing capability in IT systems. We present a case study based on requirements of the Data Protection Act and an application that audits the processing of private data, which we apply to an example manipulating private data in a university.

Keywords: Provenance, Audit, Private Data, Data Protection Act

1. INTRODUCTION

Intensive deployment of information technology and generalized online facilities by means of the Internet present amazing opportunities for businesses, governments and academic institutions such as personalisation and customisation of services, directly targeted to their audience. To strike a balance between the undoubted advantages of wider access to information and the protection of personal data, legislators have been putting legal frameworks into place, mandating institutions to address the problem of governance of their data management systems. (e.g UK Data Protection Act [8], EU [5], US Safe Harbor [22] and HIPAA [21])

Despite such regulatory frameworks, several public and highly visible cases of data leaks and exposures of private data to unintended recipients are inevitably breaking the population's trust and confidence in such new infrastructures [16, 15]. Investigations for such cases have to answer complex questions, such as establishing how an IT system allowed an individual to access, process, or communicate specific data. Hence, in light of those cases, legislators are considering more stringent powers to police institutions and enforce compliance [17].

Against such background, *auditing the use of private data* becomes a crucial capability for IT systems that brings trust and supports multiple usages. *(i)* In a forensic context, where a privacy breach is investigated, accurate auditing allows the complete case to be replayed and analysed, so as to understand the origin of the problem and its exact causes. *(ii)* In a law enforcement context, auditing offers inspection powers with which the behaviour of systems can be checked to be compliant, using exhaustive case reviews or by statistical sampling. *(iii)* From a governance viewpoint, it is good practice for institutions to set up policies and monitor their effectiveness by means of auditing capabilities. *(iv)* In the presence of outsourcing, auditing capabilities may be required by a data controller in order to audit the behaviour of a subcontractor, actions of whom it may be responsible for. *(v)* When auditing is continuous, alarms can be put in place if executions are not compatible with policies.

Understanding the causes of a breach, finding the origin of some data, checking that the processing of some data is compatible with the purpose for which the data was captured are all typical questions that an auditing capability should support. Such questions are also so-called *provenance questions* [12]. Provenance captures causal dependencies between data and events in computer systems, explaining what contributed to a piece of data in a specific state [13]. With such an explicit representation, it becomes possible to interpret and judge the quality of data, and consequently derived trust in results produced by applications.

Hence, the aim of this paper is to expose the principles by which data provenance can help audit the processing of private data in IT systems. Specifically, the contributions of this paper are: *(i)* A set of auditing use cases for private data use. *(ii)* A provenance-based architecture for auditing. *(iii)* A provenance-based application for auditing the processing of private data. The structure of this paper as follows. In Section 2, the Data Protection Act is briefly explained and three of its principles are analysed to expose their auditing requirements. In Section 3, the provenance concept and the provenance approach are explained, as well as, the methodology used for creating provenance-aware applications. In Section 4, an architecture of a provenance-based auditing system is presented to explain how provenance can help perform the auditing process. In Section 5, a provenance-based application for auditing the processing of private information based on the requirements explained in Section 2 is presented. Also, some associated results are presented and explained. Finally, Section 6 discusses some related work and Section 7 outlines future work and offers some concluding remarks.

2. CASE STUDY: DATA PROTECTION ACT

In this section, the Data Protection Act's principles related to audit the processing of private data are presented as a case study. The reason for adopting this case study is that the Data Protection Act (DPA) is the main legislative framework related to information privacy in computer systems in the UK. First we briefly introduce the DPA and, then an analysis of the selected DPA's principles is discussed to expose their auditing requirements. Finally, an example of the use of DPA is explained. Such an example will be used in the next sections to explain the audit process.

2.1. Introduction to the Data Protection Act

The DPA 1998 provides protection for an individual's personal information placing restrictions on how organisations can use personal information that they hold - including how they acquire, store, share or dispose of it. The UK's DPA is an implementation of the European directive [5] that enforces the protection of an individual's personal data as it is processed or moved between Member States of the European Union.

The DPA defines three entities that are involved in the processing of information: the *Data Controller* (DC) is the individual or organisation that decides the purpose for which, and the manner in which, personal information is to be processed; the *Data Subject* (DS) is an individual whose information is held by DC; and the *Data Processor* (DP) is any individual or organisation, other than an employee of DC, that processes personal data on behalf of DC. The information processed by these entities is classified into two types. *Personal data* (or *personal information*) is any information related to a living individual that can be used to identify that individual. *Sensitive personal data* is a special subclass of personal data that can only be processed under certain conditions. Examples of this type are: racial or ethnic origin, political opinions, religious beliefs, physical and mental condition, sexual life, offences, etc. The ways of processing personal information include obtaining, recording, holding or carrying out any operations. For collecting and processing personal data, DC has to declare a purpose: a *purpose* is the intention for which the data is to be processed. The consequence of processing personal data is called *result*. Thus, there are two types of data: *collected data* that is the information obtained from DS and *used data* that is the information used in the processing of a result. Note that the DPA was mainly created to evaluate the performance of DC and DP, thus the actions performed by the DS are not audited. Hence, the work presented in this paper is based on that assumption.

2.2. Auditing Requirements

An *audit* is an evaluation of an organization or system performed to ascertain the validity and reliability of its information and provide an assessment of its operations. The goal of an audit is to express an opinion about the organization or system under evaluation, based on accepted standards, legislative frameworks, or mutual policies. An audit is performed by an entity known as *auditor*, who then issues a report on the results of the audit. In the case of the DPA, the Information Commissioner plays the role of auditor [8] that assesses the compliance to the DPA principles.

This section analyses three of the DPA's eight principles (see [8] for the full list): Principle 2, 3 and 7, which introduce requirements for auditing private data processing. Each principle is analysed to derive its respective requirement. Such an analysis was made taking information from specialised papers [1, 3, 20, 6] and verifying their correct application with experts in the area.

Principle 2 *Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes.*

In Principle 2, the DC may only request data from DS for a legal purpose. When DS receives a DC request, DS has to verify the request and decide whether to send the required information or not. If DS decides to accept the request, then DS is giving her/his consent to process the information just for the given purpose. When DC receives the information, such information can only be processed according to the purpose agreed by DS. After processing the information, an auditor can verify that the processing made by DC or by DP (if the processing was outsourced) was really the same stated in the purpose. This provides us with an auditing requirement:

Requirement A (Purpose Verification) For verifying that DC and DP used DS's information only for the stated purpose, we require an explicit description of the processing that was applied to such information. This description has to show the processes applied to produce a result and the corresponding purposes. Thus, the description can be used by auditors to verify that the processes applied to each piece of information were compatible with the stated purposes. Moreover, auditors can verify if the DC's purposes are legal.

Principle 3 *Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.*

In Principle 3, the data requested from DS has to be relevant for the stated purpose. DC should not request, collect or use more information than required. When data processing is complete, an auditor can verify that all the data used by DC (or by DP, in the case of an outsourced processing) was relevant. Hence, we derive a new auditing requirement:

Requirement B (Relevant Information Verification) For verifying that DC and DP used only relevant information when processing data, we require a description of the information used to derive a specific result. The relevance of the information used is determined by the purpose. Then, an auditor can use the description to verify which information was used in the process of a result and decide the relevance of such information to the purpose.

Principle 7 *Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.*

According to Principle 7, DC has to offer technical and organizational measures for managing DS's data, a set of controls comprising the best practice in information security, as explained in ISO- 17799 [9]. All the organizational and administrative measures are beyond the scope of our work. We are focusing on the technical measures, specifically on the measures relating to the basic information security characteristics defined in ISO-7498-2 [10]. These are secrecy, integrity, authentication, non-repudiation and access control. Thus, auditors should be able to verify that the entities in the DPA are transporting information in a secure way, i.e. applying the basic information security characteristics to the information managed.

Requirement C (Basic Security Characteristics Verification) For verifying that DS, DC and DP are implementing the basic security characteristics in network communications, we require an explicit description of the related security process applied to data. This description can be used by auditors to verify which security characteristics have been used during the processing of information.

2.3. Application example

This section presents an example of the application of the DPA in the management of personal information by a university. The university (Data Controller) requests personal information from its students (Data Subjects) to provide an educational service (purpose). The information requested is personal details that include name, address, age, sex, nationality and school. The purpose of providing education includes "statistical processing" comprising: average of ages, average of age in each school, number of international students (by school), number of national students (by school), etc. These statistics could be calculated, among others, to support freedom of information requests, to monitor equal opportunity practices or to influence recruitment strategies. In our specific case, the university performs itself the task of calculating the statistics to a specialised external organisation (concretely, the Data Processor is the "Admissions and Students Data" unit, but in other scenarios such processing could be outsourced to a third party). Then, the age of each student is sent to such a unit that calculates their average (average of ages). The information requested by the university is considered private; therefore, the DPA should be applied in the processing of such information including the performing of audits to verify that the university really processes the requested information based on the purpose.

Note that, this example could be mapped into many different real life applications that manage personal information, e.g. government procedures, scholar registers, buying goods in Internet, facebook, etc. For more information about organisations' purposes consult the Information Commissioner's Office web page [18].

Summarising, if systems were equipped with capabilities supporting requirements A, B and C, auditors would be allowed to evaluate the actions and events related to processing of private information to ascertain the degree of correspondence between such actions and an established criteria. Thus, it would become possible to find a breach in the compliance to the established criteria. Now, performing such an audit using an automated tool would make this procedure easier, faster and more trustworthy. To do so, a description of past processing applied to data would be necessary: this is exactly what provenance technology offers us. In the next section, we introduce the concept of provenance and its principles, as an approach that allows us to implement the mentioned requirements to perform automated audits.

3. PRINCIPLES OF PROVENANCE

In Section 2, we have identified three auditing requirements derived from the DPA. These requirements have in common the need for an explicit description of past processes and the identification of the information used to generate a specific result. To provide for these needs, we advocate the use of Provenance, which we now present.

3.1. Concept of provenance

The word provenance is used in diverse areas, such art, archaeology and palaeontology, for describing the history of custody of an object since its creation, including any successive changes made to it. It is necessary to have documented evidence of such events to establish that this object has not been altered and it is not a forgery or a reproduction. The main purpose of this documentation is to prove that a specific object really comes from where it is thought to originate from, in other words, that it is authentic. This same concept of provenance also applies to data generated within computer applications to determine the origin of a computational result and its event history. Thus, provenance can be defined as: "The provenance of a piece of data is the process that led to that piece of data." [7] To support such vision, computational applications need to capture extra information that describes what actually occurred at execution time; such extra information is referred to as *process documentation*. A *provenance-aware* application can be defined as an application that besides doing the task for which it was designed, also records process documentation during its execution. Such documentation is recorded

in a storage component, *provenance store*, and queried to obtain the provenance of some data. Process documentation consists of a set of assertions, called p-assertions, asserted by the components of an application. There are two types of them: *Interaction P-Assertion* is a description of the contents of a message by an actor that has sent or received such message and *Relationship P-Assertion* is a description of how an actor obtained output data sent in an interaction, by applying some function or algorithm to input data from other interactions. [7]

The entity that takes responsibility for recording process documentation is called the *asserter*, whereas the entity issuing queries is referred to as the *querier*. *Provenance queries* are user-tailored queries over process documentation aimed to obtain the provenance of electronic data [13]. The intent of a provenance query is to select a set of p-assertions, which we refer to as *query result*, that provides the provenance of a specific piece of data. Such specific piece of data for which provenance is sought is referred to as *data item*. Not all the p-assertions related to the data item are part of a query result, just the ones that belong to a predefined context called *scope*. A query result is represented by a directed acyclic graph (DAG) that indicates where and how the data was used. Such a DAG starts with the data item followed by the relationships in scope that represent the processes that result in such data item. Thus, following relationships in a DAG helps us identify how a data item was produced. Figure 4 gives an example of such a DAG, which we will explain later in the paper. Therefore, a DAG is the representation of the provenance of a piece of data which allows users to understand what exactly happened with the data.

3.2. PrIMe

In this section, we introduce the *Provenance Incorporating Methodology* (PrIMe) [14] a software engineering methodology, which is applied at design stage, for adapting applications to make them provenance aware. PrIMe is divided into three phases: provenance question capture and analysis, actor based decomposition and adapting the application.

Phase 1 Provenance Question Capture and Analysis. In the first phase of PrIMe, an analysis of the application identifies the provenance related questions to be answered about the application. Likewise, the information for which provenance is sought is characterised as well as the answer's scope.

Phase 2 Actor Based Decomposition. In the second phase of PrIMe, the application is conceptually decomposed into a set of actors, which record process documentation. After that, their interactions are analysed to find the relationships between these interactions. This analysis exposes the information flow within the application. For this purpose, PrIMe uses a graph-based representation of application interactions. The next step is to determine which application actors are involved in the provenance of a given data item: these actors are called knowledgeable actors. If the information necessary for answering the provenance questions is available from the current actors, then it is time to apply the Phase 3. If not, Phase 2 has to be repeated until the correct level of granularity to answer provenance questions is reached.

Phase 3 Adapting the Application. This phase involves adapting the application in order to make explicit those information items that are currently implicit, hereby giving the application the necessary functionality to record process documentation, which in turn allows actors to perform queries on the documentation in order to answer provenance questions.

Equipped with this methodology, we can now fully exploit the previously explained provenance approach for developing provenance-aware applications. Thus, the PrIMe methodology will be used to provide support to the auditing requirements. In the next section, a Provenance-based Auditing Architecture based on the DPA is presented. Such an architecture includes the described provenance approach to show how it helps perform auditing processes.

4. PROVENANCE-BASED AUDITING ARCHITECTURE

This section presents a Provenance-based Auditing Architecture for auditing the processing of private data. Such an architecture was created based on the DPA case study, which was presented in Section 2. The architecture shows the entities involved in the processing of private information and their relation

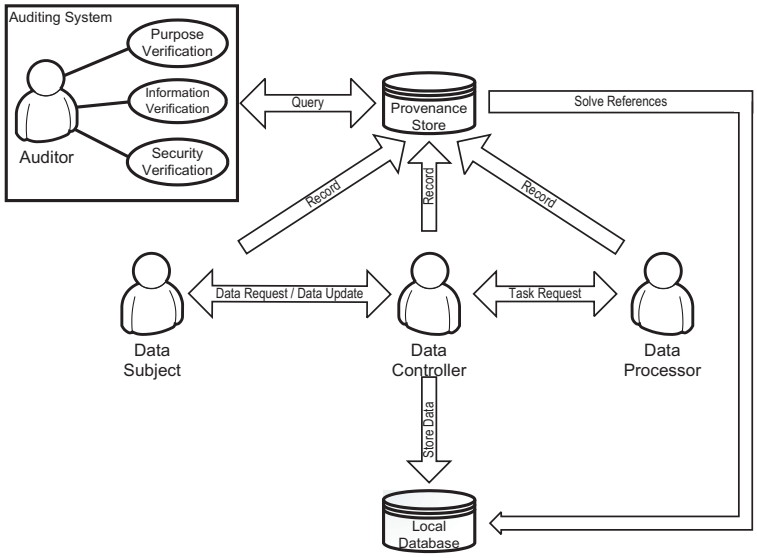

FIGURE 1: Provenance-based Auditing Architecture

with the auditing use cases. Hence, the architecture is presented as a use case diagram that shows the relationship between the entities and the auditing requirements.

Figure 1 shows the Provenance-based Auditing Architecture with four actors: `Data Subject`, `Data Controller`, `Data Processor`, `Auditor` and two databases: `Local Database` and `Provenance Store`. DS is represented by the `Data Subject` actor that is a software proxy representing a user that communicates with the `Data Controller` actor, which represents DC, through the `Data Request` and the `Data Update` processes. The `Data Request` process represents a request for personal information issued by the DC to the DS. `Data Controller` requests information from `Data Subject`, then `Data Subject` sends this information which `Data Controller` stores in a local database represented by the actor `Local Database`. The `Data Update` process represents the updating process performed by DS: if for some reason the `Data Subject` information, which is stored in `Local Database` by `Data Controller`, has changed, then `Data Subject` sends such new information to `Data Controller`. DP is represented by the `Data Processor` actor that communicates with the `Data Controller` using the `Task Request` process representing a task delegated to DP by DC. The p-assertions generated by the `Data Subject`, `Data Controller` and `Data Processor` actors during their execution are recorded in the `Provenance Store`. This process is represented by the `record` arrows that connect the actors with the `Provenance Store`. Additionally, in the Provenance Store references to the original data are recorded instead of the whole original information. Thus, `Provenance Store` has a link with `Local Database` to solve such data references that a query result has. Finally, the `Auditor` actor is a software proxy representing an auditor who wants to verify the compliance to the principles mentioned in section 2. Then `Auditor` has three auditing requirement to verify: `Purpose Verification`, `Information Verification` and `Security Verification` representing Requirement A, B and C, respectively. The three requirements can be verified by querying the p-assertions stored in the Provenance Store. By this reason, the `Auditing System` has a link with the `Provenance Store` representing the querying of p-assertions. By querying the p-assertions already recorded in the Provenance Store, the auditor can obtain a description of the process that was applied to a specific piece of information and make the necessary checks.

In this section, we have defined a Provenance-based Architecture for auditing the processing of private data based on the requirements presented in Section 2. The next step is to apply PrIMe to build a provenance-based application that implements the presented architecture and allows us to verify auditing requirements. Thus, in the next section we describe such an application that was designed using PrIMe methodology.

5. PROVENANCE-BASED APPLICATION: ANSWERING PROVENANCE QUERIES

Using PrIMe, we design a provenance-based application for auditing the processing of private data based on the example of Section 2.3 and the architecture presented in Section 4. Such an application provides support to examine the auditing requirements presented in Section 2. Due to space restrictions, in this section we only present the design obtained from the application of PrIMe and the query results obtained from the application. For clarity reasons, such queries are explained using the example of processing private information presented in Section 2.3: a university requests private information from its students for carrying out statistics.

Phase 1 Tables 1, 2, and 3 show the provenance related questions, which are derived from the auditing requirements. Each table presents a provenance question and its corresponding provenance query or queries. The data item and the scope are defined to delimit the results from the provenance query. Finally, a processing step is explained, when required to answer provenance questions.

Table 1 presents the provenance question that checks that a specific result was obtained by processing personal data that is compatible with the purpose for which it was captured. This query covers **Requirement A**, *Purpose Verification*. Table 2 presents the provenance questions that check that all the personal data captured from DS were used to obtain result. This provenance question is related to the **Requirement B**, *Relevant Information Verification*.Table 3 presents the provenance question that checks if the process to obtain a specific result received the basic security characteristics. This question is related to **Requirement C**, *Basic Security Characteristics Verification*.

TABLE 1: Provenance related question of Requirement A

Provenance Question	Verify that the result was obtained by processing personal data that is compatible with the purpose for which it was captured.
Provenance Query	What was the *used data* and the *purpose* to obtain result?
Data item	Result
Scope	Requested and collected data
Processing step	Compare used data with an established criteria related to the purpose.

TABLE 2: Provenance related question of Requirement B

Provenance Question	Was all the collected data used in the processing of result?
Provenance Query	What was the *used data* and the *collected data* used to obtain result?
Data item	Result
Scope	Requested and collected data
Processing step	Compare the used data with the collected data and highlight the differences. Any difference means data collected but not used or used but not collected.

TABLE 3: Provenance related question of Requirement C

Provenance Question	Check if the used data to obtain result was encrypted, signed,decrypted, etc.
Provenance Query	What was the *used data* and the *security processes* applied to it?
Data item	Result
Scope	Provenance of result with security processes
Processing step	Check that data was encrypted, verified, etc.

Phase 2 In the second phase of PrIMe, actor-based decomposition, resulted in the graph showed in Figure 2. This figure presents the graph of the basic communication established between the main entities of the DPA. The graph has three actors: `DC` (Data Controller), `DS` (Data Subject) and `DP` (Data Processor). Information flows in four messages denoted `M8`, `M11`, `M23` and `M26`. In `M8`, `DC` sends the purpose of collection to `DS`. Then, in `M11`, `DS` sends the required data to `DC`. These two messages are the main part of the Data Request process that represents the request of personal data. Then, in `M23`, `DC` sends a set of data (which includes the data of `M11`) that is processed by `DP`, and followed by the result to `DC` in `M26`. These messages are part of the Task Request process that represents the outsourcing of a job. Both processes can be seen in Figure 1.

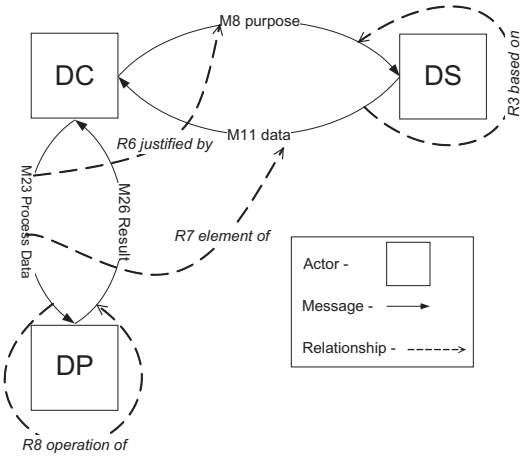

FIGURE 2: Phase 2 Graph

Figure 2 also presents the relationships between the messages, denoted R3, R6, R7 and R8: R3 indicates that M11 was sent *based on* M8, i.e. purpose; R6 indicates that M23 was sent *justified by* M8, i.e. purpose; R7 indicates that the content of M23 (Process data) has an *element of* the data in M11; and R8 indicates that the content of M26 (Result) is the result of an *operation* applied to the data in M23. Note that in Figure 2 the messages related to security do not appear in the graph, they will be included in the next phase.

Phase 3 PrIMe third phase adapts the application in order to help make explicit those information items that are currently implicit. In our case, the initial graph shown in Figure 2 does not have explicit information about the security functions applied to the data, which is necessary to answer the provenance query shown in Table 3. Therefore, the graph was modified by including a new local actor called *Security Entity* which makes explicit the information related to security. Figure 3 presents the new graph with local security entities and security functions showing the interactions and the relationships of the entire process. In this figure, the sent - received messages, by which interaction p-assertions can be inferred, are represented by solid arrows, and relationship p-assertions are represented by dashed arrows. The local security entity performs all the operations related to security (encryption, decryption, sign, verify, etc). Thus, DC, DS, and DP are each associated with a local security entity, called DCSec, DSSec and DPSec respectively.

In Figure 3 the assertions related to security functions haven been included. The protocol TLS [2] (Transport Layer Security) is used to establish communication between entities. This protocol allows the entities involved to verify each others' certificates and create a session key based on their public and private keys which are used to encrypt/decrypt every message. Moreover, the messages have the secrecy property using data encryption. Also, integrity, authentication and non-repudiation properties are provided by the use of digital signatures. These properties are achieved with the use of public and private keys. Therefore, each message is signed and verified after which it may be accepted. Access control is implemented with public certificates. Every entity has access to the granted resources after verifying its identity through its certificate. As a result, in Figure 3, M2, M5 (between DC and DS), M17 and M20 (between DC and DP) have made explicit the implementation of TLS protocol. The messages sent between entities and their corresponding local security entities represent the encryption, decryption, signing, and verification processes. Therefore, the messages sent between DC–DS and DC–DP, which are transported over a network, are encrypted and signed before being sent, and verified and decrypted after being received. Due to space restrictions, in this diagram, we do not show the processes that encrypt/decrypt and verify/sign the p-assertions. However, these security properties are applied to the p-assertions for protecting them in their transportation to the Provenance Store.

After recording interaction p-assertions and relationship p-assertions, queries were constructed based on Tables 1, 2 and 3. The results of the queries consist of DAGs displayed in figures 4 and 5. To explain

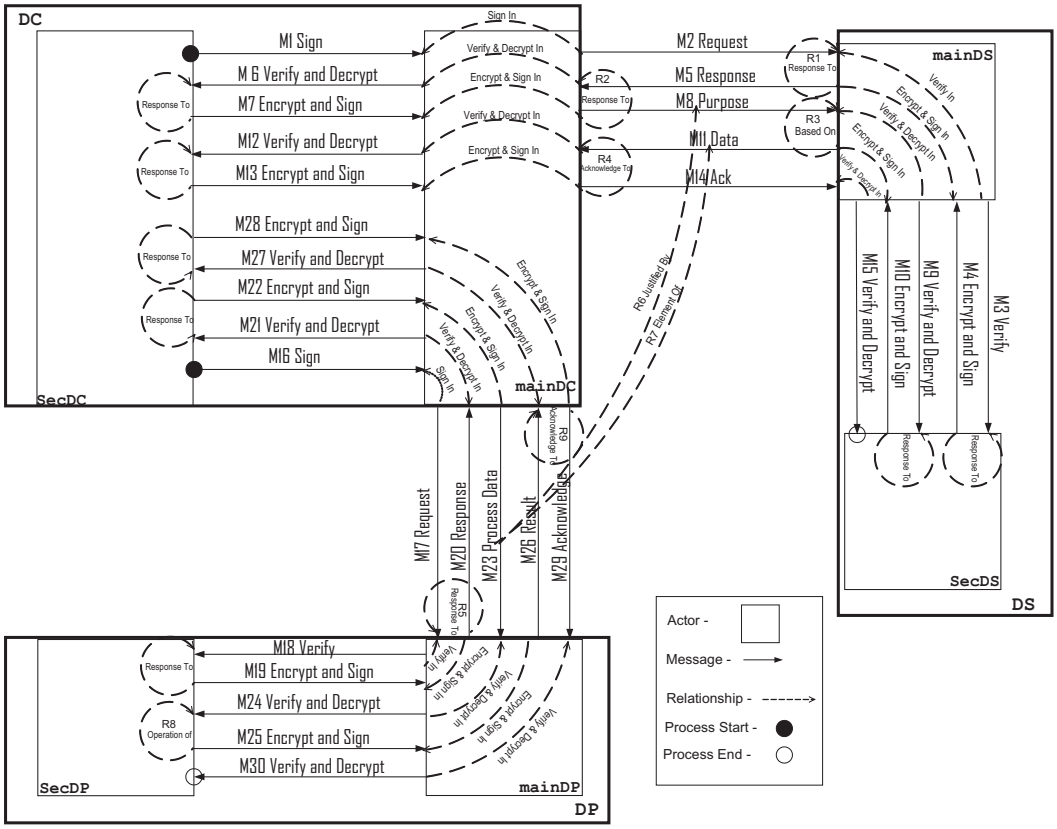

FIGURE 3: Phase 3 Graph

how to audit the processing of private data using such queries, we use the example presented in Section 2.3.

To answer the provenance questions in Table 1 and Table 2, it is necessary to identify two kinds of data: *used data* and *collected data*. Collected data is the information sent by DS to DC (M11), as a result of request (M2). Used data is the information sent by DC to DP that is contained in M23 and used to obtain a result (M26), which is the consequence of processing the requested data. Note that used data is a subset of collected data and message M23 could contain more information than such used data (e.g. instruction of processing, information necessary for performing the processing). Therefore, such extra information is not private and is out of the auditing process.

To answer these questions, the assertions related to security are not necessary. Therefore, a provenance query should obtain a DAG of *Result without* security operations. Figure 4 shows such a DAG: at its right, we display the piece of data whose provenance is sought, in this case *averageAge*. Relationships by which this result was obtained are shown in bold, whereas the data related with each relationship is shown in normal text. Used data is identified in the DAG with a solid oval and collected data with a dashed oval. The message number (related to Figure 3) in which the data is transported is denoted by M*i*.

For answering Table 1 question, we need to identify the used data to obtain *averageAge*, which is *age1*, and the purpose, which is "statistical processing", which includes "averages of ages". If we follow the arrows in Figure 4, we can see that *averageAge* is an **average of** ages (age1, age2, age3), which is the used data, and in the case of "age1", this operation was **justified by** the purpose "statistical processing". Then we can affirm that *averageAge* was processed following Principle 2.

For answering Table 2 question, we need to identify the used data to obtain *averageAge*, which is *age1*, and the collected data, which is name, age, nationality, school. If we compare them, then we can see that they are different. The reason is that the purpose "statistical processing" includes different tasks

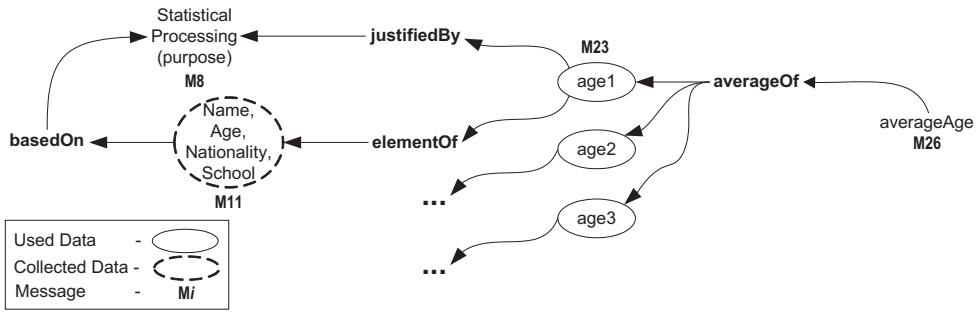

FIGURE 4: Provenance DAG of *averageAge* **without** security operations

besides "average of ages" (e.g. the average of age in each school, the number of international students, etc). Therefore, name, nationality, and school have to be processed in accordance to the established purpose, "statistical processing", to be in compliance with Principle 3. For example, to use *nationality* for processing the number of international students or *school* for processing the average age in each school.

To answer the provenance question in Table 3 it is necessary to show the provenance of *averageAge with* security operations, as in Figure 5. In Figure 5, the encrypted data is marked with a parallelogram, the plain data with a rectangle and the message number (related to Figure 3) in which the data is transported is denoted by M*i*.

The encryption/signing relationship is denoted **encSignIn** and the verification/ decryption relationship is denoted **verDecIn**. Thus, when a data is signed and encrypted before being sent and an entity receives such data (cryptodata), the entity has to verify the sign and decrypt the data to obtain and process the plain data in some way. Thus, if the DAG of a data has the relationships **encSignIn** and **verDecIn**, then such data was transported using a complete digital signature scheme. In this way, the audit can ascertain that some data offers security characteristics (in this case: secrecy, integrity, authentication, and non-repudiation), as required by Requirement C.

In Figure 5, the data item is *cryptoaverageAge*, which is the same data item as the showed in Figure 4 but signed and encrypted. If we follow the arrows in the figure, we can see that *cryptoaverageAge* is the result of the encryption/signing of *plainaverageAge*, which is the **averageOf** *plainAge*. And *plainAge* is the result of the verification/decryption of *cryptoage1*. Therefore, in Figure 5 *cryptoaverageAge* was processed using information that was transported using a digital signature scheme. Then, we can affirm that *cryptoaverageAge* was processed following Principle 7.

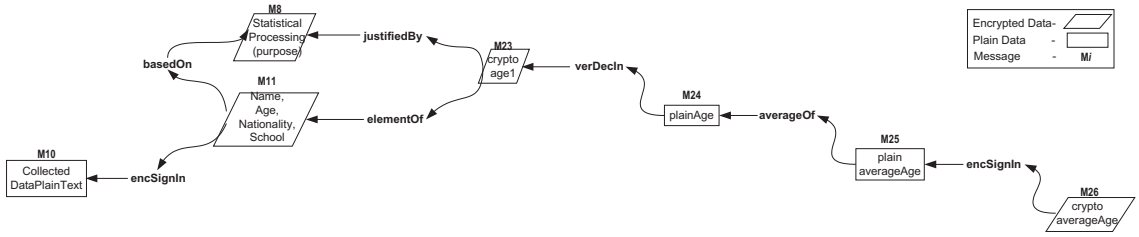

FIGURE 5: Provenance DAG of *cryptoaverageAge* **with** security operations

To summarise, we have presented the design and the results of a provenance-based application for auditing the processing of private data that allow us to verify if the processing of information is in compliance to a legislative framework. We have also made explicit security characteristics in the processing of data to cover the presented auditing requirements. Moreover, including security characteristics to the application allows auditors to trust in the retrieved information during the verification of the processing of private information.

6. RELATED WORK

The traditional way of auditing includes a qualified auditor who assesses the degree of compliance of a pre-defined set of specifications. This assessment is a "hand-made" process that follows certain procedures established by standards. These procedures are long and, because of the human intervention, prone to errors. In order to increase efficiency and to allow for continuous auditing, many researchers [19, 4, 11, 23] have proposed to automate the auditing processes using different technologies.

Philip et al [19] and Chorley et al [4] propose the idea of using provenance to create audit reports related to Evidence-Based Policy Assessment (EBPA). They suggest using provenance to evaluate the quality and reliability of data, the robustness of an analysis, and the validity of findings in an assessment. However, they do not present any analysis that shows how to use provenance in this specific field.

Kifor et al [11] proposes the use of provenance in electronic healthcare record systems. The authors also investigate the privacy risk aspects of introducing provenance into healthcare systems. They propose not to store sensitive medical data in the provenance store, but only references, to avoid unauthorized users being able to link some piece of medical data with an identifiable person. The described technique is already applied in our work, not only for security reasons but also for scalability. The approach of using provenance information for making audits is mentioned but not fully discussed. They neither present any audit use case nor present an analysis of the security characteristics. They mention the HIPAA regulation, however, our design can be applied to such a regulation.

Xie et al [23] proposes the use of a probabilistic integrity audit method to verify query integrity in outsourced databases. The authors present the use of this method to verify the correctness and completeness of queries in outsourced databases reducing costly requirements of sending back query results. This method could complement our work by using it to verify the correctness and completeness of the queries in the provenance store. However, we are focusing on how to audit the use of private data in an automatic way and the security characteristics that we include in the application allows us to have a certain level of trust in the audit results.

7. FUTURE WORK AND CONCLUSION

Our work demonstrates that the use of provenance technologies provides an appropriate tool for an automated verification of the compliance not only of the DPA, but also of other legislative frameworks related to processing of private information (e.g. [22],[21]). Our aim is to provide an easy way to audit the correct use of private personal information in distributed systems. Hence, this research defines a Provenance-Based Architecture for Auditing and presents a provenance-based application for auditing processing of private data. Thus, part of our future work is to evaluate the presented architecture and application to strengthen our proposal. As well, part of this work is to develop a provenance-based generic middleware to audit the processing of private information that can be used in applications independently of platforms and programming languages. On the other hand, raising the level of trust in the query result can be achieved by creating a formalisation of the presented architecture. Besides, to make use of ontologies to define purposes, collect data, used data, etc. in the DAGs can help perform an automatic analysis of such DAGs.

In summary, this document describes the application of provenance concepts to audit private data processing in distributed applications. It presents the Data Protection Act as a case study and a requirement analysis of three of its principles that are related to auditing procedures. These requirements were modelled for presenting a provenance-based architecture and showing how provenance can help the auditing process. A provenance-based application was developed using the presented provenance approach and some associated query results, which are in the form of a DAG, were presented. These results were analysed to show that provenance can be used as a means for auditing the correct processing of private information. Thus, the presented provenance-based application provides a means for verifying that organizations are in compliance with legislative frameworks related to processing private information.

REFERENCES

[1] David Bainbridge. Data protection ı- data protection, the new law. *Computer Law & Security Report*, 14 no. 3:180–184, 1998.

[2] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, and T. Wright. Transport Layer Security (TLS) Extensions. *RFC 3546*, June 2003.

[3] Peter Carey. *Data Protection - A practical Guide to UK and EU Law*. Oxford, 2nd edition, 2004.

[4] Alison Chorley, Pete Edwards, Alun Preece, and John Farrington. Tools for tracing evidence in social science. In *Proceedings of the Third International Conference on eSocial Science*, October 2007.

[5] European Comission. Directive 95/46/EC of the European Parliament, October 1995.

[6] Council for Science and Technology. Better use of personal information: opportunities and risks. Technical report, Council for Science and Technology, 2005.

[7] Paul Groth, Sheng Jiang, Simon Miles, Steve Munroe, Victor Tan, Sofia Tsasakou, and Luc Moreau. D3.1.1: An architecture for provenance systems. Technical report, University of Southampton, February 2006.

[8] HomeOffice. Data Protection Act, 1998.

[9] ISO-17799. ISO/IEC 17799:2005 Information Technology - Security Techniques - Code of Practice for Information Security Management.

[10] ISO-7498-2. ISO 7498-2:1989 Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.

[11] Tamás Kifor, László Varga, Sergio Álvarez, Javier Vázquez-Salceda, and Steven Willmott. Privacy issues of provenance in electronic healthcare record systems. In *Journal of Autonomic and Trusted Computing (JoATC)*, volume issue 3, 2008.

[12] Simon Miles. Electronically querying for the provenance of entities. In *Proceedings of the International Provenance and Annotation Workshop IPAW*, pages 184–192. Springer, November 2006.

[13] Luc Moreau, Paul Groth, Simon Miles, Javier Vázquez, John Ibbotson, Sheng Jiang, Steve Munroe, Omer Rana, Andreas Schreiber, Victor Tan, and László Varga. The provenance of electronic data. *Communications of the ACM*, April 2008.

[14] Steve Munroe, Simon Miles, Luc Moreau, and Javier Vázquez-Salceda. PrIMe: A Software Engineering Methodology for Developing Provenance-Aware Applications. In *Proceedings of Sixth International Workshop on Software Engineering and Middleware SEM 06*, Portland, Oregon, 2006.

[15] BBC News. Data lost by revenue and customs. http://news.bbc.co.uk/1/hi/uk/7103911.stm, 21 November 2007.

[16] BBC News. UK's families put on fraud alert. http://news.bbc.co.uk/1/hi/uk_politics/7103566.stm, 20 November 2007.

[17] House of Commons Justice Committee. Protection of private data – first report of session 2007–08. Technical report.

[18] Information Comissioner's Office. Data protection register. http://www.ico.gov.uk/.

[19] Lorna Philip, Alison Chorley, John Farrington, and Pete Edwards. Data Provenance, Evidence-Based Policy Assessment, and e-Social Science. In *Proceedings of the Third International Conference on eSocial Science*, 2007.

[20] Chris Pounder. Security and the new data protection law. In *Computers & Security*, volume 17, pages 124–128, 1998.

[21] United States Public Law. Health insurance portability and accountability act of 1996. 104-191, 1996.

[22] US-Commerce-Department. Safe harbor, 2000.

[23] Min Xie, Haixun Wang, Jian Yin, and Xiaofeng Meng. Integrity auditing of outsourced data. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 782–793, 2007.