Inferring Accountability from Trust Perceptions

Koen Decroix, Denis Butin, Joachim Jansen, Vincent Naessens

▶ **To cite this version:**

Koen Decroix, Denis Butin, Joachim Jansen, Vincent Naessens. Inferring Accountability from Trust Perceptions. Information Systems Security - ICISS 2014, Dec 2014, IDRBT, Hyderabad, India. pp.69 - 88, 2014, Lecture Notes in Computer Science. . .

HAL Id: hal-01118593

https://hal.inria.fr/hal-01118593

Submitted on 19 Feb 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Inferring Accountability from Trust Perceptions

Koen Decroix[1], Denis Butin[2], Joachim Jansen[3], and Vincent Naessens[1]

[1] KU Leuven, Technology Campus Ghent, Department of Computer Science
Gebroeders Desmetstraat 1, 9000 Ghent, Belgium
{koen.decroix,vincent.naessens}@cs.kuleuven.be
[2] Inria, Université de Lyon, France
denis.butin@inria.fr
[3] Department of Computer Science, KU Leuven, Belgium
joachim.jansen@cs.kuleuven.be

Abstract. Opaque communications between groups of data processors leave individuals out of touch with the circulation and use of their personal information. Empowering individuals in this regard requires supplying them — or auditors on their behalf — with clear data handling guarantees. We introduce an inference model providing individuals with global (organization-wide) accountability guarantees which take into account user expectations and varying levels of usage evidence, such as data handling logs. Our model is implemented in the IDP knowledge base system and demonstrated with the scenario of a surveillance infrastructure used by a railroad company. We show that it is flexible enough to be adapted to any use case involving communicating stakeholders for which a trust hierarchy is defined. Via auditors acting for them, individuals can obtain global accountability guarantees, providing them with a trust-dependent synthesis of declared and proven data handling practices for an entire organization.

Keywords: Accountability, IDP, Trust, Privacy, Surveillance

1 Context and Motivation

Contemporary situations involving the exchange of personal data for services often leave individuals oblivious as to the actual processing of their data. While privacy policies are widely used by organizations across the world, they often constitute mere declarations of intent. Individuals generally cannot check whether actual processing is in line with such ex ante statements. Furthermore, privacy policies often remain purposely vague while users demand concrete promises about the retention of their data, the purposes for which it is used, obligations in terms of third party forwarding and so on.

The rise of individuals' expectations about data handling transparency, combined with the growing imbalance of power between them and data processing organizations, has made the principle of *accountability* a key component of the discourse over privacy protection. While the concept of accountability was already mentioned in this context in the eighties [24], it appears more prominently

these days. In particular, the upcoming European General Data Protection Regulation [12] cites accountability explicitly. Organizations will therefore increasingly be legally required to be accountable for their data handling practices to data subjects.

A downside of this concept's popularity is that its meaning has been diluted due to frequent use in different contexts. Lawyers often focus on procedural aspects of accountability [7, 25]. Computer scientists often tackle specific security properties — such as non-repudiation [2] — or specific technical contexts like cloud environments [15]. Because of these varied interpretations, no universal definition of accountability can be given. However, it normally refers to the necessity of surpassing mere compliance to achieve demonstration of compliance. By putting the burden of proof of good behavior on the data controller, accountability measures increase pressure on organizations to be transparent and fair in their data handling practices.

In real-world situations, data shared by an individual does often not remain within the realm of a single entity but it disseminated among communicating subsystems that may even be geographically distant. Since subcontractors may belong to different corporations than the organization that collected personal data in the first place, different data handling policies may apply. The initial data controller may fulfill its promises as long as data remains in its initial location, but offer no guarantees about processing by other stakeholders. Such situations leave individuals blind to the whereabouts of their data.

Even if all involved entities publish clear data handling policies, the end result is opaque to individuals. Technical privacy policies may be very detailed and the number of entities may be large. It is useful for individuals to understand the resulting global (organization- or system-wide) guarantees that apply to their personal data. If individuals have defined personal privacy preferences for themselves once and for all, they would also like to know whether the overall processing of their data by an organization and its subcontractors is in conflict with those preferences.

This paper introduces a model capable of inferring global accountability guarantees from the point of view of a trusted auditor. This auditor acts on behalf of the user and represents his interests. In practice, the auditor could be a member of a Data Protection Authority or a third-party, accredited auditing organization. The framework allows the hierarchical representation of entities in an organization, thereby modeling trust relationships: an individual may only trust a given component in an organization, or may trust an entity higher in the hierarchy, thereby trusting all components operating by that entity. These trust assumptions (i.e. user expectations) influence the computation of the global accountability guarantees. We distinguish between three levels of users: a naive user, a regular one and a privacy-aware one. The level of privacy-awareness of a user influences the kind of evidence this user assumes to be trustworthy.

In addition to these different types of users, data handling statements carry different levels of evidence. Each entity subcontracting for an organization has its own data handling statements. At the lowest level, statements are merely

declarations of intent with no additional evidence. This level of evidence is akin to a detailed, technical privacy policy. Other statements are provided together with system traces of data handling operations, i.e. logs. These logs are assumed to be trustworthy, but they have not been inspected. Therefore, it may not be obvious at first glance that a data processor has misbehaved, even though a trace of misbehavior is assumed to exist in the logs. The situation where logs cannot be checked easily is realistic because logs are not standardized in general, many organizations use very specific formats and because semantics are often unavailable. The highest level of evidence features statements that are accompanied by logs that have been verified and found to be compliant. Here, it is again assumed that the logs are trustworthy, i.e. reflect actual system execution, and that the log analysis software is sound and accurate.

The three levels of user privacy-awareness and three levels of statement evidence are combined to compute fine-grained global accountability guarantees. The auditor, on behalf of the user, can both inspect those global guarantees or detect potential conflicts by providing the privacy preferences of the individual.

Our framework is implemented in IDP, a knowledge base system [11]. We demonstrate it through the scenario of a surveillance infrastructure managed by a railroad company and involving a third-party security service company, operators such as a surveillance guard and an image processor, and components used by these operators. This kind of scenario demonstrates the typical situation where an individual shares his data with only one entity initially, after which the entity processes and disseminates the data among several subcontractors. Assuming individuals are monitored via cameras, one can distinguish between several categories of personal data which can be collected, processed and distributed. Depending on image quality and on pan-tilt-zoom functionality, cameras may record full body pictures with insufficient quality to distinguish faces, full body pictures with blurred faces, faces only or even record behavior patterns while discarding body images.

As mentioned above, we assume logs (when they exist) to be trustworthy: they are accurate and cannot be forged by entities. In practice, this requires techniques such as forward integrity [3] to guarantee the security of logs, and partial formal modeling or trusted computing to ensure unforgeability. While these criteria are important, they are outside of the scope of this work: here, we suppose that logs reflect actual system execution and therefore embody meaningful evidence. Furthermore, we presume that personal data is categorized in a standardized way, so that individuals and organizations use the same terminology for categories of personal data.

We continue with some technical background on the IDP system, a knowledge base system based on an extension of typed first-order logic (§2). The approach is illustrated by the running example of a railroad surveillance infrastructure, presented informally at first (§3). We then introduce the building blocks of the accountability inference framework and apply it to this scenario (§4). After evaluating the results of this implementation of the model (§5), we discuss related work on formalizations of accountability and privacy (§6), including ex-

isting models for privacy reasoning realized with IDP. The paper concludes with a discussion of the potential, limitations and future of the framework (§7).

2 IDP

IDP [16, 29] is a state-of-the-art knowledge base system [8] developed by the Knowledge Representation and Reasoning (KRR) group at KU Leuven. We briefly introduce IDP and how it can be used as a tool to manage an accountability framework, focusing on the parts of the system relevant for this paper. More interested readers can find IDP documentation and source code here [16], and some examples here [17]. In this text we use IDP to refer to IDP^3, the current version of the IDP system. One of the main focuses of IDP is knowledge representation: allowing users to formulate their knowledge in a intuitive manner. To this end the FO($\cdot$) language framework, an extension of First Order Logic (FO), was developed. Using this language, users can model their data (in this case, which organizations or data categories to analyze), as well their knowledge (here, accountability across organizations) in a formal way using logical formulas (constraints) and definitions. This model can be used to solve problems by applying one of the many *inferences* IDP provides. For this paper we will need *(optimal) model expansion*: given a partial assignment for data, find a complete (optimal) assignment such that all expressed constraints and definitions hold. The initial, partial assignment corresponds to the setting of our framework: a hierarchical network of organizations and the accountability guarantees they offer. The outcome of the model expansion inference then corresponds to a complete assignment: a listing of which information is used in what places and what kinds of accountability guarantees it offers. This will later be called the Global Accountability Profile (GAP).

There exists a variety of declarative modeling systems, such as Alloy [19, 20] or ASP solvers [13,22]. We chose to use IDP as our modeling tool for two reasons. First, the language it uses is expressive and intuitive: it supports extended first order constraints as well as definitions under well-founded semantics [26]. Second, it is implemented as an extension in Lua [18], which means there is support for procedural integration. This allows us to determine the way in which we want to use our declarative model in a flexible way.

3 A Railway Station Surveillance Scenario

To illustrate the model, we consider the scenario of video camera surveillance in a railway station. Since individuals are filmed by cameras, the collected categories of personal data are related to images (we assume the cameras do not record sound). Several categories of personal data can be inferred from camera recordings, such as identification through face detection [27], gait recognition [21], behavioral tracking [23] and many others. Signs inform passersby that the *Railway Company* installed *Cameras* for video surveillance. The cameras provide the railway's *Monitors* in the control room with real-time video feeds containing

Blurred Faces and *Gaits* of travelers. Furthermore, detailed images of individuals' *Full Body* and *Gait* are stored in the railway's *Image Database* serving as *Evidence* in legal investigations. Only authorized *Image Processors* employed by the railway company have access to it. Additionally, surveillance *Guards* employed by a third-party *Security Company* patrol in the station. They are authorized to view real-time images on the monitors, and carry a *Mobile Device* for registering *Contextual Data* (e.g. time and location) in case of incidents. These devices are connected with the *Status Database*, property of the security company. It is only accessible for the security company's *Status Processors* upon request of legal institutions for collecting *Evidence*.

The trusted auditor (acting on behalf of a filmed individual) is external to the model and we focus on data handling statements from the entities collecting personal data, listed in Tab. 1.

4 Components of the Accountability Inference Model

Having set the stage for both our model and the tool that will be used to evaluate it, we now describe the framework's building blocks (depicted in Fig. 1) in detail. Entities, all related to a core organization, provide individual statements about their data handling practices. These statements can be provided together with unverified logs, verified logs, or exist on their own without companion evidence. Different categories of personal data can be modeled. As a consequence, statements are fine-grained enough to express different guarantees about various types of personal data. The data subject is represented by a trusted auditor. This auditor takes into account the subject's trust perceptions. Global accountability guarantees are automatically computed using the computation rules in the *System Independent Part* of the framework. These guarantees are represented by the *Global Accountability Profile* (GAP) that is automatically inferred, using a *Knowledge Base System* (IDP), from the *System Model* and the *User Model*, both part of the framework's *Input Model*. The former models the individual statements, the relations between the entities expressing the statements, and the level of evidence characterizing the statements. The latter includes the level of trust of the user. As a consequence, the global accountability guarantees take into account both factual evidence and subjective appreciations of privacy risks. This combination reflects the fact that different data subjects demand different levels of proof to be satisfied. The model provides data subjects with an overview of the accountability guarantees resulting from a set of interacting entities. In addition, it allows them (or the auditor, on their behalf) to check whether their personal privacy preferences are compatible with this global accountability panorama. The remainder of this section further details the framework's elements.

4.1 Personal Data

Organizations collect personal data of data subjects that interact with systems owned by these organizations. Being accountable to data subjects involves clarifying which types of their personal data are harvested and used. These categories

Table 1. Camera surveillance data handling statements. Entity statements are (D)eclarative, (L)ogged-unverified or Logged-and-(V)erified.

(R)ailway Company, (C)amera, (M)onitor, (I)mage Database Statements		
Stat R.1	(L)	Full body pictures with blurred or clear faces, gaits, heights, and behavior are recorded for incident detection.
Stat R.2	(D)	Collected pictures containing evidence of incidents can be forwarded to legal authorities upon their request.
Stat R.3	(L)	Pictures are never collected for commercial purposes.
Stat R.4	(L)	The maximal retention time for any category of collected personal data is 60 days.
Stat C.1	(L)	Cameras in the station record full body pictures with blurred or clear faces, gaits, heights, and behaviors of travelers for incident detection purposes.
Stat M.1	(L)	Guards monitor in real-time full body pictures with blurred faces, gaits, heights, and behaviors of travelers in the station for incident detection purposes.
Stat I.1	(L)	Full body pictures with clear faces are stored as evidence of possible incidents.
Stat I.2	(V)	Access to stored full body pictures with clear faces is only granted to the image processor upon request of the legal authorities.
Stat I.3	(V)	Full body pictures with clear faces, gaits, heights, and behavior are never processed for the purpose of identification.
Stat I.4	(D)	Stored images are deleted after 30 days, unless they are being used as evidence in legal cases.
(S)ecurity Company, M(O)bile Device, Status (D)atabase Statements		
Stat S.1	(D)	Time and location of incidents are collected as evidence.
Stat S.2	(L)	Time and location of incidents are only forwarded to legal authorities upon request.
Stat O.1	(V)	Surveillance guards collect time and location as evidence in case of incidents.
Stat D.1	(V)	Time and location of incidents are collected as evidence.
Stat D.2	(V)	Access to stored time and location of incidents is granted to status processors for gathering evidence.
Stat D.3	(V)	The time and location of incidents are deleted after 90 days unless they are being used as legal evidence.

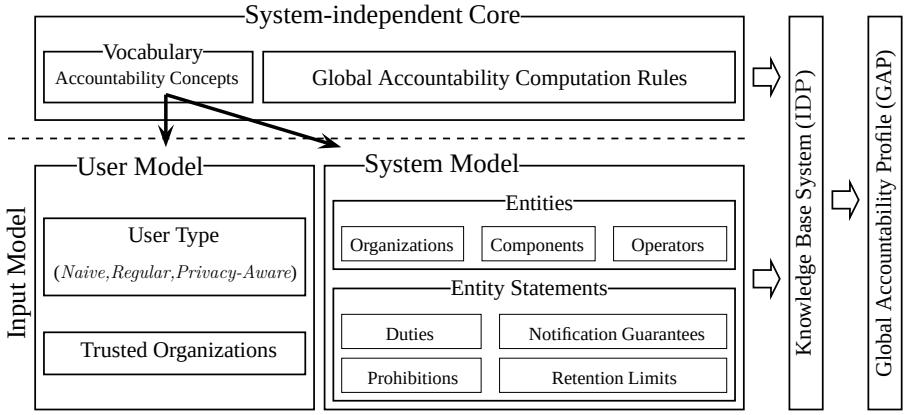

Fig. 1. Structure of the global accountability inference model.

are represented by the *DataCategory* type. All categories of collected personal data involved in a given scenario must be spelled out in the input model as the contents of *DataCategory*.

One can define hierarchies of personal data categories. This models the fact that categories of personal data can be subsets of other categories, e.g. the age of an individual gives strictly more information than a predicate on whether the individual is over 18. The data category hierarchy is represented using *DataCategoryOf(DataCategory,DataCategory)*, which deduces hierarchical knowledge from the initial specifications. Listing 1 depicts the IDP input model for the personal data and their hierarchy of the camera surveillance scenario.

```
type DataCategory = {  PersData; Face; BlurredFace; Gait; Height; Behavior;
        Location; Time, PictureIncident  }
DataCategoryOf(DataCategory, DataCategory) = {
        Face, PictureIncident; BlurredFace, PictureIncident; Gait,
        PictureIncident; Height, PictureIncident; Behavior, PictureIncident  }
```

IDP Listing 1: Partial user model representing personal data categories and hierarchies in the video surveillance scenario.

4.2 Entities

Data subjects and the auditors that act on their behalf are not explicitly modeled since their point of view is external. An arbitrary number of active entities can be modeled in the framework's system model. Active entities are those that handle personal data of the subjects and provide some degree of accountability,

i.e. declarations (with or without proof) about the data processing they perform. A distinction is made between *Stakeholders* and *Components*. A stakeholder is either an *Organization*, or an *Operator* acting on behalf of exactly one organization. An organization can employ more than one operator. Components are constituents of data processing systems. A component belongs to exactly one organization, but can be used by multiple operators.

Components process personal data under the responsibility of the organizations that own them. Organizations or authorities may restrict access to the data categories that a given component is capable of collecting. Authorized categories for a given component are specified using *ComponentCanCollect(Component, DataCategory)*. Listing 2 depicts the IDP specification of the entities involved in the camera surveillance scenario.

```
type Entity = { RailwayCompany; SecurityCompany; LegalAuthority; Camera;
        Monitor; MobileDevice; SurveilanceGuard; ImageProcessor;
        StatusProcessor; ImageDB; StatusDB }
type Stakeholder isa Entity = { RailwayCompany; SecurityCompany;
        LegalAuthority; SurveilanceGuard; ImageProcessor; StatusProcessor }
type Component isa Entity = { Camera; Monitor; MobileDevice; ImageDB;
        StatusDB }
type Organization isa Stakeholder = { RailwayCompany; SecurityCompany;
        LegalAuthority }
type Operator isa Stakeholder = { SurveilanceGuard; ImageProcessor;
        StatusProcessor }
ComponentOf(Component) : Organization = { Camera → RailwayCompany;
        Monitor → RailwayCompany; ImageDB → RailwayCompany;
        StatusDB → SecurityCompany; MobileDevice → SecurityCompany
EmployeeOf(Operator) : Organization = {
        SurveilanceGuard → SecurityCompany;
        StatusProcessor → SecurityCompany;
        ImageProcessor → RailwayCompany }
OperatorOf(Operator, Component) = { SurveilanceGuard, Monitor;
        SurveilanceGuard, MobileDevice; ImageProcessor, ImageDB;
        StatusProcessor, StatusDB }
ComponentCanCollect(Component, DataCategory) = { Camera, Face;
        Camera, BlurredFace; Camera, PictureIncident; Camera, Gait;
        Camera, Height; Camera, Behavior; Monitor, Face; Monitor,
        BlurredFace; Monitor, PictureIncident; Monitor, Gait; Monitor, Height;
        Monitor, Behavior; ImageDB, Face; ImageDB, BlurredFace;
        ImageDB, PictureIncident; ImageDB, Gait; ImageDB, Height;
        ImageDB, Behavior; ImageDB, Time; ImageDB, Location; StatusDB,
        Time; StatusDB, Location; MobileDevice, Time; MobileDevice,
        Location }
```

IDP Listing 2: Partial system model representing the entities and their relationships in the video surveillance scenario.

4.3 Statements and Local Accountability Statements

All entities involved in data handling relevant to a given data subject are assumed to exhibit some level of accountability of practice, i.e. they publish precise declarations about their intended personal data handling practices. In general, each entity publishes a different data handling statement. A one-to-one mapping between entities and data handling statements is assumed, and is modeled using function *StatementFrom(Statement) : Entity*. Listing 3 shows the part of the system model that defines a subset of the *statements* of the *railway company* listed in Tab. 1 [4]. Those statements include the following aspects:

- Purposes of use, i.e. the list of finalities for which the collected personal data may be used (for instance statistics or direct marketing) — this is modeled using *StatementPurpose(Statement, Purpose)*. Multiple purposes can be defined for a statement.
- The category of personal data that is used, i.e. the collection of personal identifiable information. Possibly, multiple subject data categories exist for a statement — this is modeled using predicate *StatementSubject(Statement, DataCategory)*.
- Global retention limits, i.e. the period of time after which the personal data will be deleted by the entity (e.g. 30 days) — this limit is expressed using a partial function (i.e. not every statement expresses a retention limit) *StatementRetentionLimit(Statement) : Duration*.
- *Obligations* built from a *Condition* and an *Action*. These are modeled using partial functions *StatementCondtion(Statement) : Condition* and *StatementAction(Statement) : Action*). Both are partial functions because not all statements are linked to actions (e.g. retention limits), and unconditional obligations are modeled by only modeling the actions of statements.
- Personal data may be forwarded to organizations. In the model, this is expressed using *StatementDestination(Statement, Organization)*. Possibly, a statement has multiple destinations.

Obligations are flexible and can be used to express a variety of constraints. Conditions are events that trigger a reaction, e.g. the personal data is accessed or the data subject has requested an update. Actions are the resulting events, for instance the update of his personal data or its forwarding.

Statements guaranteeing the sending of a notification (to a user) when a specific event occurs (e.g. when a specific category of personal data is accessed by the entity) are expressed using *StatementNotificationGuarantee(Statement)*.

Accountability occurs at different levels. Some entities may merely declare their intended practices, without providing any companion evidence. Other entities provide data handling logs. In the model this is denoted using function *StatementProof(Statement) : StatementEvidence*. It may not always be possible to check the compliance of data handling logs with obligations. Logs can be in a

[4] For the complete model of the statements, see `https://code.google.com/p/inferring-accountability/`.

format which is not standardized, or semantics may not be provided by the entity. We therefore distinguish between three levels of assurance (*StatementEvidence*) for data handling statements:

1. A statement is (purely) *Declarative* if data handling logs relevant to the statement are not made available by the entity publishing the statement.
2. If data handling logs are provided together with the statement but cannot be checked straight away, the statement obtains the status *LoggedUnverified*.
3. If a statement is provided together with logs that have been checked for compliance (e.g. through a trusted log analysis software), the statement is said to be *LoggedVerified*. This is the highest level of accountability for a data handling statement, since actual behavior has both been recorded and shown to be compliant with the statement.

4.4 Trust Perception and Global Accountability Inference

While organizations may feature complex hierarchies with heterogeneous data handling practices, individuals care about what happens to their personal data globally. A panoramic overview of the worst-case scenario in terms of data processing (i.e. *what are the weakest global guarantees?*) is relevant to individuals, since they must often decide whether to interact with an entire organization. Most of the time, they cannot cherry-pick with which subcontractors to share their data with.

Global accountability inference is a central feature of this framework that builds such a synthetic statement for data subjects. It deduces global guarantees from the local accountability statements of all entities involved in the system. These (subjective) guarantees depend on trust perceptions of data subjects.

Individuals display different levels of trust in the entities that handle their personal data. The framework's user model reflects this socio-technical aspect by modeling three levels of trust, corresponding to three typical types of individuals:

- *Naive* individuals always trust data handling statements, even if statements are purely declarative (i.e. no evidence in the form of a log is provided);
- *Regular* individuals only trust statements co-occurring with relevant data handling logs;
- *Privacy-aware* individuals are most skeptical and trust only statements for which verified logs have been provided by issuing entities.

Furthermore, the user model includes *UserTrust(Organization)*, the user's high-level trust perception towards organizations. It represents his trust in declared data handling practices of related organizations. This also implies that all operators they employ and components they own are trusted by him. The modeled video surveillance scenario features the aforementioned three user models: naive (*U1*), regular (*U2*) and privacy-aware (*U3*). It also assumes that no organization is trusted, i.e. *UserTrust(Organization)* is the empty set.

```
type Statement = { StatR1; StatR2; StatR3; StatR4; ... }
type Purpose = { Evidence; DetectIncident; Commerce; Identification }
type Condition = { RequestLegalAuthority; NoLegalInvestigation }
type Action = { Collecting; Monitoring; Storing; Forwarding; Accessing }
type Duration isa int = { 30; 60; 90 }
type Permission constructed from { Always; Never }
type StatementEvidence constructed from { Declarative;
        LoggedUnverified; LoggedVerified }
StatementFrom(Statement) : Entity = { StatR1 → RailwayCompany;
        StatR2 → RailwayCompany; StatR3 → RailwayCompany;
        StatR4 → RailwayCompany; ... }
StatementSubject(Statement, DataCategory) = { StatR1, Face;
        StatR1, BlurredFace; StatR1, Gait; StatR1, Height; StatR1, Behavior;
        StatR2, PictureIncident; StatR3, PictureIncident; StatR4, PersData;
        ... }
StatementPurpose(Statement, Purpose) = { StatR1, DetectIncident; StatR2,
        Evidence; StatR3, Commerce; ... }
partial StatementCondtion(Statement) : Condition = {
        StatR2 → RequestLegalAuthority; ... }
StatementPermission(Statement) : Permission = { StatR1 → Always;
        StatR2 → Always; StatR3 → Never; StatR4 → Always; ... }
partial StatementAction(Statement) : Action = { StatR1 → Collecting;
        StatR2 → Forwarding; StatR3 → Collecting; ... }
StatementDestination(Statement, Organization) = {
        StatR2, LegalAuthority; ... }
partial StatementRetentionLimit(Statement) : Duration = {
        StatR4 → 60; ... }
StatementNotificationGuarantee(Statement) = { }
StatementProof(Statement) : StatementEvidence = {
        StatR1 → LoggedUnverified; StatR2 → Declarative;
        StatR3 → LoggedUnverified; StatR4 → LoggedUnverified; ... }
```

IDP Listing 3: Partial system model representing the statements of the entities involved in the video surveillance scenario.

This impact of these user trust models on the perception of global accountability guarantees is shown in Tab. 2. For instance, a *naive user* considers he is guaranteed that merely declared statements of an entity E, owned or employed by organization O, correspond with actual data handling practices. By contrast, a *regular user* only considers merely declared statements to be guaranteed if he trusts O (i.e. *UserTrust(O)*), and assumes statements provided together with logs to be guaranteed, whether these logs are checked for compliance or not.

Table 2. Global statement evidence deduction rules — the global evidence for the statement S by the entity E owned by the organization O is *(U)ncertain* or *(G)uaranteed* for the modeled user.

StatementProof(S)=	Declared	Logged-unverified	Logged-and-verified
Naive user	G	G	G
Regular user	$F(E) : \{G, U\}^\star$	G	G
Privacy-aware user	$F(E) : \{G, U\}^\star$	$F(E) : \{G, U\}^\star$	G

$^\star F(E) =' G' \Leftrightarrow \mathit{UserTrust}(O) \wedge (\mathit{ComponentOf}(E) = O \vee \mathit{EmployeeOf}(E) = O)$
$^\star F(E) =' U' \Leftrightarrow \neg \mathit{UserTrust}(O) \wedge (\mathit{ComponentOf}(E) = O \vee \mathit{EmployeeOf}(E) = O)$

Global statement computations are performed differently for *duties* (i.e. statements featuring *Always*) and for *prohibitions* (i.e. statements featuring *Never*). Beside these categories, models also include statements expressing *notification guarantees* and *global retention limits*. Comparable with duties, these also feature *Always*. Nevertheless, these are treated differently in computations.

Global statements are expressed in terms of *global data categories* (i.e. users are concerned what happens to their personal data). Let S be an individual statement of entity E, *CanCollect(E,DC)* a data category DC that can be collected by an entity E, and *Sub(S,DC)* representing that data category DC is a subject of S. Tab. 3 summarizes the worst-case deduction rules that depend on the *global statement evidence* for the computation of *GlobalDataCategory(S,DC)*, the global data categories derived from S.

Duties. Global statements using *Always* are built as follows:

- The *global purposes of use* for a global data category are constructed from the union of all purposes of (individual) duties S, with *GlobalDataCategory(S,DC)*. These represent worst-case global purposes which are conjunctive. For instance, personal data is collected for commercial and statistical reasons. If no purpose is explicitly specified, then all purposes are assumed to be permitted globally.
- The *global conditions of use* for a data category are constructed from the disjunction of all conditions of duties S, with *GlobalDataCategory(S,DC)*. If

Table 3. Worst-case computation rules for deducing *GlobalDataCategory(S,DC)*, the global data categories DC deduced from the individual statement S of entity E, with *Sub(S,DC)* the subject DC of statement S, and $CanCollect(E, DC)$ the data categories collectable by E.

Global statement evidence of S:	**Uncertain**	**Guaranteed**
$Duty(S)$	$CanCollect(E, DC)$	$\psi(S, E, DC)^\star$
$Prohibition(S)$	$\psi(S, E, DC)^\star$	$Sub(S, DC)$
$NotificationGuarantee(S)$	$Sub(S, DC)$	$\psi(S, E, DC)^\star$
$RetentionLimit(S)$	$Sub(S, DC)$	$\psi(S, E, DC)^\star$

$^\star\psi(S, E, DC) \equiv CanCollect(E, DC) \wedge Sub(S, DC)$

at least one unconditional statement exists, no overall conditional statement is generated.
- The *global actions* for a data category are built from the union of all actions of individual duties S, with *GlobalDataCategory(S,DC)*.
- The *global level of assurance* (i.e. global evidence) for a data category is *Uncertain* if at least one uncertain statement (in the sense of Tab. 3) exists for this data category. Else, the global statement is considered *Guaranteed*.
- The *global notification guarantee* for events relative to a data category is built from the conjunction of all individual notification guarantees S relative to that data category, with *GlobalDataCategory(S,DC)*.
- The *global retention limit* for a data category is the maximum of all retention limts S existing for the data category, with *GlobalDataCategory(S,DC)*.

Prohibitions. Global statements using *Never* are built as follows:

- The *global purposes of use* for a global data category are constructed from the union of all purposes of individual prohibitions S, with *GlobalDataCategory(S,DC)*. These represent worst-case global purposes which are disjunctive. For instance, personal data is never collected for commercial or statistical reasons. Individual prohibitions without explicit purpose are omitted during the deduction of global purposes (i.e. worst-case).
- The *global conditions of use* for a data category are constructed from the conjunction of all conditions of prohibitions S, with *GlobalDataCategory(S,DC)*. Unconditional statements are omitted.
- The *global actions* for a data category is computed as for duties, *mutatis mutandis*.
- The *global level of assurance* for a data category is computed as for duties, *mutatis mutandis*.

In global statements, trust is expressed in a binary way (i.e. *GAPEvidence*): statements are, from the point of view of the data subject, either *Uncertain* or *Guaranteed*. Both global notification guarantees and global retention limits, part of the GAP, are expressed as duties. Global statements present guarantees

as a function of categories of personal data. Once global statements have been computed, they are represented as in Listing 4. They are categorized as follows:

- *Global duties about collecting personal data* — declaring actions about the use, collection, or storage of personal data.
- *Global duties about distributing personal data* — declaring actions that forward data to external organizations.
- *Global prohibitions for collecting personal data* — expressing that the use, collection, or storage of personal data is forbidden.
- *Global prohibitions for distributing personal data* — forbidding the forwarding of personal data to an organization.
- *Global notification guarantees* — declaring the sending of a notification upon the occurrence of a specific event.
- *Global retention limits* — expressing the time limit after which all categories of personal data must be deleted.

5 Computation and Evaluation

We illustrate a possible use of the framework with an IDP realization [1]. Our realization infers the GAPs of the user models *U1*, *U2*, and *U3* described earlier, representing individuals under video surveillance in a railway station. The model was generated in less than a second on a personal computer. We first compare the resulting profiles for naive, regular, and privacy-aware data subjects. Next, we discuss how the statements of entities and users are modeled.

5.1 Trust-Dependent GAP Inference

First, given the type of user and his trust perception toward organizations, we deduce for each entity the user's global statement evidence using the rules of Tab. 2. For instance, *U2* (i.e. regular user) is sufficiently guaranteed that data practices comply with declared ones if they are merely logged. Instead, *U3* is satisfied when statements are *logged and verified* by an auditor or just logged in case organizations are trusted by him. This global evidence is then used for the deduction of the GAP using the rules of Tab. 3. The inferred GAPs are summarized in Tab. 4. None of these contain global prohibitions. However, individual statements of entities include two prohibitions (i.e. *R.3* and *I.3*). The reason for this is that worst-case computation rules give priority to global duties containing data categories that are subject of both duties and prohibitions. Semantically, this corresponds to a user who is more concerned about the categories of data used rather than about the unused ones.

[1] A detailed IDP realization — together with the output containing the GAPs for the three user models — can be found at `https://code.google.com/p/inferring-accountability/`.

```
type GAPEvidence constructed from { Uncertain; Guaranteed }

GAPCollectData(DataCategory)
GAPCollectDataAction(DataCategory, Action)
GAPCollectDataForPurposeOf(DataCategory, Purpose)
GAPCollectDataCondition(DataCategory, Condition)
GAPCollectDataProof(DataCategory, GAPEvidence)

GAPForwardDataTo(DataCategory, Organization)
GAPForwardDataAction(DataCategory, Action)
GAPForwardDataForPurposeOf(DataCategory, Purpose)
GAPForwardDataCondition(DataCategory, Condition)
GAPForwardDataProof(DataCategory, GAPEvidence)

GAPNeverCollectData(DataCategory)
GAPNeverCollectDataForPurposeOf(DataCategory, Purpose)
GAPNeverCollectDataCondition(DataCategory, Condition)
GAPNeverCollectDataProof(DataCategory, GAPEvidence)

GAPNeverForwardDataTo(DataCategory, Organization)
GAPNeverForwardDataForPurposeOf(DataCategory, Purpose)
GAPNeverForwardDataCondition(DataCategory, Condition)
GAPNeverForwardDataProof(DataCategory, GAPEvidence)

GAPNotificationGuarantee(DataCategory)
GAPNotificationGuaranteeCondition(DataCategory, Condition)
GAPNotificationGuaranteeProof(DataCategory, GAPEvidence)

GAPRetentionLimit(DataCategory, Duration)
GAPRetentionLimitCondition(DataCategory, Condition)
GAPRetentionLimitProof(DataCategory, GAPEvidence)
```

IDP Listing 4: Modeling concepts representing the GAP.

Global duties for collecting data are perceived differently by *U1*, *U2*, and *U3*. Since *U1* is satisfied with statements that are purely declarative, he believes that data collection duties are respected by the organizations. Having the same guarantees *U2* is only partially convinced, since he requires at least data handling logs while the security company's duty *S.1* is purely declarative. Therefore, *Time* and *Location*, subjects of *S.1*, are considered as global duty data categories that are uncertain. *U3* needs the strongest guarantees. He is not convinced for any of the data categories part of the GAP. He expects for all data collection duties that logs exist and that they are verified by an auditor. For instance, because duty *R.1* is *logged-unverified*, the duty subjects, such as *Face* and *BlurredFace*, are not sufficiently guaranteed for *U3*. Furthermore, due to *R.1* an additional data category *PictureIncident* is deduced in *U3*'s GAP. This fol-

lows from the computation rule in Tab. 3 for duties with global evidence that is uncertain for *U3*, and the given railway station's camera capability *ComponentCanCollect(Camera,PictureIncident)*. Also, comparing the GAP of *U3* with the others, more purposes for collecting data are deduced. In particular, besides that *BlurredFace* and *Gait* are collected for incident detection (i.e. *DetectIncident*), these are used as *Evidence* of incidents as well.

Global data forward duties are computed from the declarative duty *R.2* and the merely logged duty *S.2*. Both duties declare to forward data to other stakeholders in the system. The results show that *U1* is satisfied with the guarantees provided that the system respects data forwarding declarations. The same guarantees are too weak to convince *U2* and *U3*. Both doubt that actual data practices correspond with declared ones. At first glance, one could expect that *U2* assumes that *Time* and *Location*, part of the GAP, are used as declared by *S.2*. However, *S.2* is redundant with the purely declared duty *R.2* because *Time* and *Location* are subjects of *R.2* as well. This can be deduced using the computation rules of Tab. 3 and from the railway company's image database capabilities, namely *ComponentCanCollect(ImageDB,Time)* and *ComponentCanCollect(ImageDB,Location)*.

Global retention limits are computed from *R.4*, *I.4*, and *D.3*. Results show that retention limits are conditional (i.e. *NoLegalInvestigation*) for all data categories in the GAP of *U1* and *U2*. The GAP of *U3* shows an additional unconditional retention limit for data category *PersData* (i.e. personal data). This is deduced from *R.4*, which provided evidence not fulfilling *U3*'s expectations. Indeed, *R.4* is just logged, and not verified. Furthermore, *U2* only has partial guarantees for *Time* and *Location* since evidence of *R.3* sufficiently guarantees him. In case of *U3*, *R.3* provides insufficient evidence. Hence, usage of *Time* and *Location* are not sufficiently guaranteed according to *U3*'s GAP.

5.2 Statements Modeling and User Models

The statements presented in the scenario are atomic declarations, i.e. they consist of single actions on subject data categories. Concepts in the framework were defined for modeling atomic statements. However, statements may contain multiple declarations. The concepts defined lack expressiveness for modeling these. These statements must be represented by the atomic parts from which they are composed. For instance, the image database is associated with a declarative statement announcing the storage of personal data of categories *blurred face* and *gait* for a maximum of 30 days and for the purpose of statistics and marketing. This is modeled as (*a*) a *duty* declaring that data categories *blurred face* and *gait* are *stored*, and (*b*) a *retention limit* specifying that data is kept for a maximum of 30 days. In the model, these items correspond to separate elements of the *Statement* domain. Though both statements have the same purposes, these must be expressed separately. In particular, *StatementPurpose(Statement,Purpose)* relates the purposes *statistics* and *marketing* to the *duty* and *retention limit* with *2 statements* $\times$ *2 purposes* relations. Decomposing combined statements may imply that statement relations grow combinatorially. Similarly, each *Statement*

Table 4. Inferred GAP synthesizing global accountability in the camera surveillance system for user models *U1* (1), *U2* (2), and *U3* (3). The numbers in the table indicate the user models for which the statements in the left column, represented relatively to the different data categories, are valid.

	Picture	*Incident Face*	*BlurredFace*	*Gait*	*Height*	*Behavior*	*Time*	*Location*	*PersData*
Global Collection duties									
Actions									
Collecting	3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	
Accessing		1,2,3					1,2,3	1,2,3	
Storing	3	1,2,3	3	3	3	3	1,2,3	1,2,3	
Monitoring	3	3	1,2,3	1,2,3	1,2,3	1,2,3	3	3	
Purposes									
All		1,2,3							
DetectIncident	3		1,2,3	1,2,3	1,2,3	1,2,3	3	3	
Evidence	3		3	3	3	3	1,2,3	1,2,3	
Conditions									
Unconditional	3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	
Guaranteed		1,2	1,2	1,2	1,2	1,2	1	1	
Global Forward duties									
Actions									
Forwarding									
to *LegalAuthority*	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	
Purposes									
Evidence	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	
Conditions									
RequestLegalAuthority	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	
Guaranteed	1	1	1	1	1	1	1	1	
Global Retention Limits									
Duration (days)									
60 days	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3			3
90 days							1,2	1,2	
Conditions									
UnConditional									3
NoLegalInvestigation	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	
Guaranteed	1	1	1	1	1	1	1,2	1,2	

element must be related to the *Entity "ImageDatabase"*, the *Permission "Always"*, and to the *StatementEvidence "Declarative"*. Furthermore, each duty is related to the *Action "Store"*.

Users model. The coarse-grained user categorization we use facilitates user modeling since modelers only need to specify user types via a single constant, for instance $TypeOfUser = NaiveUser$. The model's user types intuitively represent typical real-world users, determining how data subjects appreciate statements and evidence from organizations. They reflect the fact that skeptical users are more difficult to convince of the compliance of actual data handling with declared practices. The user model also addresses the high-level trust perception of users. Namely, $UserTrust(O)$ expresses that a user trusts the organization O.

Reusing framework components. The framework consists of modular components, making possible isolated changes to one part while leaving the others intact. A given system model (e.g. the railway station camera surveillance sce-

nario) is unaffected when new types of users are introduced. Similarly, if an auditor collects different samples of statement evidence, only changes to the evidence in the statement model are required.

Detecting conflicts. The user model could be extended with user privacy preferences containing prohibitions. This aspect could be used by auditors wanting to verify e.g. whether a system, run by a commercial organization, is not collecting sensitive health information. The individual statements of system entities are another flexible facet. Typically, these statements form a large set of opaque and potentially inconsistent declarations. Automated verification can be added to the system-independent part of the framework for easy conflict detection.

6 Related Work

A privacy evaluation framework based on trust assumptions is introduced in [9]. Like our model, it involves multiple stakeholders. This framework was later implemented in IDP [10]. A distinction is made between storage-trusted and distribution-trusted organizations. The privacy analysis focuses on which data is needed for access to services, and how personal data is distributed between interacting services. By contrast, this paper's model targets interactions between organizations, not services, and investigates how statements about personal data handling, backed with varying levels of evidence, combine with trust perceptions to yield assumptions about the processing of personal data.

The approach of using standardized privacy policies to enable accountability by clarifying obligations is widespread. In particular, the idea of combining privacy policies with data handling logs to automatically check compliance ex post appears in [28]. The question of the gap between system event logs and logs at the level of abstraction of privacy policies is addressed in [5]. The consequences of log design choices for log analysis and accountability are addressed in [4]. Adequate log design for compliance checking is tricky because of the numerous possible semantic ambiguities. Both papers presume a single data controller rather than the setting of this paper — a constellation of interacting data processors with different, potentially incompatible privacy policies.

Beyond computer science, the scope of application of accountability is a vividly debated issue in the privacy regulation debate [14]. A key question related to our work is how far data controllers should be required to go to demonstrate compliance. Distinctions are sometimes [6] made between different levels of accountability, ranging from public declarations of intent to full technical transparency, such as the one that we advocate here. The adequacy of procedures, i.e. organizational measures, is often discussed. Privacy Impact Assessments are often advocated [30] and can be seen as a bridge between accountability of procedures and accountability of practice if the assessment is conducted in sufficient detail. The question of privacy-preserving surveillance infrastructures is addressed in particular in the PARIS project [1], with an interdisciplinary angle.

7 Conclusions

We described an accountability inference model and its realization in the IDP knowledge base system. Trust perceptions are taken into account to compute global accountability statements from the individual statements made by interacting entities. We distinguish between different levels of proof for the individual statements, again influencing the resulting global accountability statements. Our approach is illustrated with a scenario involving stakeholders in a railway surveillance infrastructure. The framework is not tied to any particular scenario and can be extended easily. Our representation of data handling evidence is only implicit, and therefore coarse-grained. A more refined approach would model the semantics of log compliance explicitly. This level of detail seems difficult to implement within a first-order logic-based framework. In the current version of the framework, the auditor acting on behalf of an individual is not notified of privacy policy conflicts automatically. Including this aspect would remove the need for manual compatibility checking.

Acknowledgement This work was funded by the Flemish agency for Innovation by Science and Technology (IWT), the European project PARIS / FP7-SEC-2012-1 and the Inria Project Lab CAPPRIS.

References

1. PrivAcy pReserving Infrastructure for Surveillance (PARIS). `http://www.paris-project.org`
2. Bella, G., Paulson, L.C.: Accountability Protocols: Formalized and Verified. ACM Trans. Inf. Syst. Secur. 9(2), 138–161 (2006)
3. Bellare, M., Yee, B.S.: Forward Integrity for Secure Audit Logs. Tech. rep., University of California at San Diego (1997)
4. Butin, D., Chicote, M., Le Métayer, D.: Log Design for Accountability. In: 2013 IEEE Security & Privacy Workshop on Data Usage Management. pp. 1–7. IEEE Computer Society (2013)
5. Butin, D., Le Métayer, D.: Log Analysis for Data Protection Accountability. In: Jones, C., Pihlajasaari, P., Sun, J. (eds.) FM 2014: Formal Methods, Lecture Notes in Computer Science, vol. 8442, pp. 163–178. Springer (2014)
6. Colin J. Bennett: Implementing Privacy Codes of Practice. Canadian Standards Association (1995)
7. De Hert, P.: Accountability and System Responsibility: New Concepts in Data Protection Law and Human Rights Law. In: Managing Privacy through Accountability. pp. 193–232. Palgrave Macmillan (2012)
8. De Pooter, S., Wittocx, J., Denecker, M.: A Prototype of a Knowledge-based Programming Environment. In: Proceedings of the 19th International Conference on Applications of Declarative Programming and Knowledge Management (INAP 2011). pp. 191–196 (2011)
9. Decroix, K., Lapon, J., Decker, B., Naessens, V.: A Formal Approach for Inspecting Privacy and Trust in Advanced Electronic Services. In: Jürjens, J., Livshits, B., Scandariato, R. (eds.) Engineering Secure Software and Systems, Lecture Notes in Computer Science, vol. 7781, pp. 155–170. Springer (2013)

10. Decroix, K., Lapon, J., Decker, B., Naessens, V.: A Framework for Formal Reasoning about Privacy Properties Based on Trust Relationships in Complex Electronic Services. In: Bagchi, A., Ray, I. (eds.) Information Systems Security, Lecture Notes in Computer Science, vol. 8303, pp. 106–120. Springer (2013)
11. Denecker, M.: A Knowledge Base System Project for FO(·). In: Hill, P., Warren, D. (eds.) Logic Programming, Lecture Notes in Computer Science, vol. 5649, p. 22. Springer (2009)
12. European Commission: Regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), inofficial consolidated version after LIBE committee vote (2013)
13. Gebser, M., Kaufmann, B., Schaub, T.: Conflict-Driven Answer Set Solving: From Theory to Practice. Artif. Intell. 187, 52–89 (2012)
14. Guagnin, D., Hempel, L., Ilten, C.: Managing Privacy Through Accountability. Palgrave Macmillan (2012)
15. Haeberlen, A.: A Case for the Accountable Cloud. Operating Systems Review 44(2), 52–57 (2010)
16. The IDP system. `http://dtai.cs.kuleuven.be/krr/idp` (2014)
17. KRR Software: IDP examples. `http://dtai.cs.kuleuven.be/krr/software/idp-examples` (2014)
18. Ierusalimschy, R., de Figueiredo, L.H., Celes, W.: Lua – an extensible extension language. Software: Practice and Experience 26(6), 635–652 (1996)
19. Jackson, D.: Alloy: A Lightweight Object Modelling Notation. ACM Transactions on Software Engineering and Methodology (TOSEM'02) 11(2), 256–290 (2002)
20. Jackson, D.: alloy: a language & tool for relational models. `http://alloy.mit.edu/alloy/` (2012)
21. Lee, L., Grimson, W.E.L.: Gait Analysis for Recognition and Classification. In: IEEE International Conference on Automatic Face and Gesture Recognition. pp. 148–155 (2002)
22. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. ACM Trans. Comput. Log. 7(3), 499–562 (2006)
23. Mecocci, A., Pannozzo, M., Fumarola, A.: Automatic detection of anomalous behavioural events for advanced real-time video surveillance. In: IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications (CIMSA'03). pp. 187–192 (2003)
24. Organisation for Economic Co-operation and Development: OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data (1980)
25. Raab, C.: The Meaning of 'Accountability' in the Information Privacy Context. In: Managing Privacy through Accountability. pp. 15–32. Palgrave Macmillan (2012)
26. Van Gelder, A., Ross, K.A., Schlipf, J.S.: The Well-Founded Semantics for General Logic Programs. Journal of the ACM 38(3), 620–650 (1991)
27. Viola, P., Jones, M.: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154 (2004)
28. Weitzner, D.J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., Sussman, G.J.: Information accountability. Commun. ACM 51(6), 82–87 (2008)
29. Wittocx, J., Mariën, M., Denecker, M.: The IDP system: A model expansion system for an extension of classical logic. In: Denecker, M. (ed.) Proceedings of the 2nd Workshop on Logic and Search, Logic and Search. pp. 153–165. ACCO (2008)
30. Wright, D., de Hert, P.: Introduction to Privacy Impact Assessment. In: Wright, D., Hert, P. (eds.) Privacy Impact Assessment, pp. 3–32. Springer (2012)