Towards a Logic of Consent and Revocation

Ioannis Agrafiotis, Sadie Creese, Michael Goldsmith, Nick Papanikolaou
e-Security Group, WMG
University of Warwick
Coventry, England
Email: {I.Agrafiotis, S.Creese, M.H.Goldsmith}@warwick.ac.uk

***Abstract*—Our aim is to provide a mechanism for bridging the gap between data privacy policy languages and high-level requirements. We introduce a logic for reasoning about the dynamics of privacy. In particular, we focus on the semantics of the processes of consent and revocation when applied to the handling and use of personal data. Our logic provides the basis for a formal verification framework for privacy and identity management systems. It is independent of any particular policy description language for privacy preferences and privacy-aware access control, and can be used to verify correctness of policy against requirements specifications, as well as consistency across a policy set. We give examples of how the logic can be used to specify aspects of high-level privacy policies.**

CONTENTS - REMOVE THIS

I Introduction 1

II Consent and Revocation Processes 2

III Defining a Hoare Logic for Consent and Revocation 3

IV Resolving Ambiguities and Limitations in the Logic 5

V Defining an Access Control Model for Consent and Revocation Processes 5

VI Linking the Two Models 7

VII Applications 7

VIII Conclusions and Future Work 9

References 9

I. INTRODUCTION

Our society has become significantly dependent on the Internet for a range of functions which pervade our lives including citizen services, commerce, information communications and socialising to name but a few. We face a growing threat of malicious activity in cyberspace, and the public is increasingly aware of its perils. Among the most pernicious problems is that of controlling the dissemination of personal data. Since individuals are constantly required to supply personal data online, there is an acute need to implement practical control measures. The objective of our research is to develop a rigorous framework within which we can prove properties about information systems in which the privacy of individuals' personal data must be enforced.

We are reminded on an almost daily basis [1] that adequate data protection is often lacking, leading to the possibility of massive data losses from commercial organisations and government departments alike. The protection of personal data privacy typically will require data owners to take certain measures to protect themselves, when data owners are often ill equipped to do so [2]. It is therefore unsurprising that we are witnessing growing amounts of identity theft online [3]. The general consequences of loss of privacy for an individual have been studied at length from a legal perspective [4], and their exacerbation due to the huge computational resources available in current enterprises poses significant problems. Indeed, modern companies have at their disposal enormous storage and processing capabilities (cf. data mining), making personal data the object of detailed scrutiny and a significant source of value for service operators.

We aim to provide a method which will enable data controllers, the providers of services handling individuals' personal data, to more easily mitigate the risks associated with the release, handling and dissemination of personal data across information networks. Specifically, our logic will underpin a mechanism by which high-level data privacy requirements, expressed in a manner meaningful to individuals (data subjects), and policy level descriptions, expressed in a manner meaningful to the architects of data services, might be compared; such comparisons would provide assurance that a policy does indeed meet individuals' requirements, and that the various policy-level statements are mutually consistent.

We define *Consent* to be a privacy preference when applied to personal data; the act of giving consent represents a wish for personal data to be collected, or processed, or disseminated, for a particular purpose. We will consider the different types of consent in Section II. On the other hand, *Revocation* is any process which corresponds to a withdrawal of consent; it is a wish for personal data to cease to be collected, processed, or disseminated, for a particular purpose.

Technical Approach.: We first present a novel access control model suitable for expressing privacy preferences; in

this model, consent and revocation are perceived as dynamic modifications of those preferences. We feel that while this model immediately supports policy enforcement architectures such as the one being developed within EnCoRe, it does not provide an intuitive language for data subjects to express their consent and revocation behaviour.

Our second formal model, which is a simple Hoare logic, provides a core set of consent and revocation actions axiomatised in their effect on rights and permissions in a way that is more familiar to data subjects.

It is our intention to link the two models together by providing a mechanism to interpret the logic over the access control model. This will enables us to prove correctness properties. The overlap of notation between the two formalisms is not accidental.

Related Work.: There is a general lack of work specifically addressing the processes of consent and revocation in the context of personal data. While the concept of consent has been studied extensively [5] in the social sciences, leading to work on the necessity, meaning and consequences of *informed consent* [6], computer scientists have not given the mechanics of such processes due attention. Revocation is usually understood by those involved in information systems implementation as the process of invalidating a security certificate. Further, our ongoing interactions with various data subject groups indicates that revocation is not a concept they actually use. The touch point appears to be a notion of *deletion*, often interpreted as complete removal of data from a network or system. There appears to be limited insight into how else revocation might be achieved, or the technical reality of deletion [7] when applied to data stored in information systems.

Logics for reasoning about security protocols exist, and most notably in the setting of authentication [8], [9]. Logics for access control have been studied before [10]. The enforcement of privacy policies and privacy preferences may well be understood as a form of *privacy aware* access control, and so it is necessary to acknowledge here this vital theoretical connection. A variation of role-based access control which accounts for privacy, the Privacy RBAC model, has been proposed [11]. A formal framework for privacy preferences, SecPALP, has been developed by Becker et al. [12]. We note also here the work on formal comparison of privacy policies by May et al. [13], and also the 'formal privacy system' model by Gunter et al. [14]. However, none of the above specifically handle consent and revocation.

The P3P specification [15] is relevant to our work as it defines an XML schema for privacy policies and preferences (known as APPEL). We envisage being able to reason formally about rules expressed in these languages by mapping them into our logic, a topic for future research. It should be noted that P3P has been the object of debate, since no formal semantics for it exists and there are some known ambiguities [16]. However, as noted by one of its authors [17], P3P was the first substantial effort to produce a common language for privacy, and thus serves as a foundation for most of the research in this space.

Previous Work.: The research described here has been carried out within the context of the UK-wide collaborative project EnCoRe sponsored by the UK EPSRC, ESRC and TSB (see http://www.encore-project.info). The basic setting for consent and revocation management is described in [18]. In previous work, we have developed a model of revocation for personal data [7] and a CSP/FDR verification framework for validating P3P policies [19]. We have also begun addressing the gap between the levels of abstraction of privacy policies and corresponding specification languages [20].

II. Consent and Revocation Processes

The issues we are interested in reasoning about arise as a consequence of the sharing of personal data. Enterprises store personal data about their customers, often not only contact details but various consumer preferences; such information enables an enterprise to tailor its products and services to customer needs. Companies share, buy and sell customer data with the aim of increasing their customer base, and obtaining useful marketing statistics. There are even firms whose entire business is to manage and market large databases of personal data about individuals. The central problem is that an individual for whom personal data is held can barely, if at all, control (view, modify, remove) the storage, aggregation and flow of this data.

While it is necessary, by law, to obtain an individual's consent for the collection of data regarding his or her person, the possible semantic interpretations of such consent can vary widely, possibly opening the way for data misuse and abuse; this can arise when the data collector interprets said consent in a way that conflicts with the individual's preferences. Furthermore, it is common practice for an enterprise to request 'blanket consent,' which is consent for data to be collected, used and disseminated in any way the data collector deems appropriate; by giving blanket consent, an individual completely relinquishes control over such data. When more fine–grained consent is required, the request is usually accompanied by lengthy details of the collector's privacy policy; research has suggested [21] that individuals rarely study such policies carefully, and this may cause them to give blanket consent to save time and effort.

Control over personal data held regarding an individual, from that individual's point of view can be understood as the ability to *revoke*, either the data, or certain permissions to use and disseminate the data, or both. Consequently, revocation has many different flavours, with subtle differences depending on how the data and the associated consent must be altered.

The duality of consent and revocation does not always involve a symmetry: there exist scenarios in which consent for data to be collected or used has not been explicitly given, and yet an individual has the right to perform revocation. However,

the mechanism for exercising that right may not be readily (if at all) available. Similarly, there are cases in which, once consent has been given, revocation may not be allowed: this is true, for instance, in the case of profiles submitted to national DNA databases in the UK.

Types of Consent.: There are three main tasks for which a data collector requires consent from an individual:

- *collection* of personal data (for storage in a database)
- *use* of personal data (for analysis, marketing or one of many other purposes)
- sharing or *dissemination* of personal data (to the public domain, to another data collector)

Each of these cases gives rise to interesting variations and corresponding challenges. Collection can be performed in many ways (directly, indirectly), through a variety of media (an explicit registration or consent form, email, online purchases), into various forms of storage (a local enterprise server, distributed/cloud-based storage). Depending on the multiple jurisdictions within which data handling, processing and storage takes place, different restrictions and privacy legislation may apply: no single privacy policy would suffice.

The purposes for which consent is requested are practically impossible to enumerate, and no list could be exhaustive; in the P3P language, a predefined set of purposes of data use is provided. This is the aspect of consent which is hardest to pin down; even if a purpose for the use of data is unambiguously defined, it is not evident how one can check that the actual use of the data matches that purpose. To illustrate the issue, consider the case of an individual giving consent for his personal data, including health records, to be used by an enterprise for medical research. The scope of 'medical research' as the purpose of data collection is too broad for any realistic control to be applied to the data. The individual in question may be happy to have her data used for breast cancer research but not for diabetes research, as this may reveal private family history. In any case, there is no universal language for defining purposes clearly and unambiguously, making this aspect of consent difficult to quantify.

Dissemination of personal data between enterprises could cause a multitude of privacy problems, and consent for such onward sharing needs to be clearly defined and carefully enforced. An enterprise may require the services of a third party to fulfil its business needs, and in doing so share its customer database. It is up to the enterprise to ensure the third party adheres to an adequate privacy policy, and in some cases there may be cause to be even more stringent – e.g. to prevent the third party from sharing the data onward to other parties. There are other complications also: a public sector body may need to outsource data to a private enterprise, but may be bound by tighter controls since it is meant to serve the public interest; in this case if data is to be shared it may have to be thoroughly anonymised, for example.

Consent Variables.: While consent is chiefly characterised by the elements we have identified above, there are further subtleties that need to be considered. In giving consent for the collection, use and dissemination of personal data, an individual is likely to wish to impose some additional constraints on the following quantities, which we term *consent variables*:

- t: duration of consent (time-out)
- v: volume of data held
- s: sensitivity of data held
- Π: which parties may access the data
- a: persistence - how data is treated after consent has lapsed

Example. Suppose an individual wishes to impose the following constraints on his or her consent: the consent is granted for 30 days (after which period, the data must be erased and consent requested anew if necessary), the consent is valid for data that is not sensitive (the Data Protection in the UK defines precisely which types of data are deemed 'sensitive'). We might write these constraints as a simple propositional formula:

$$(t \leq 30) \wedge (s = \texttt{NONSENSITIVE}) \wedge (a = \texttt{delete}) \quad (1)$$

where the variables t, s, a are associated with a specific system model, and `NONSENSITIVE` and `delete` are suitably defined constants. It is worth noting that the constraint on a is what is known in the literature as an *obligation*, in the sense that it prescribes an action that a data controller is obligated to perform at some future time.

Types of Revocation.: The variants of revocation, as identified in [7], are the following:

- deletion (of data and permissions),
- revocation of permissions to process data,
- revocation of permissions for third party dissemination,
- revocation of identity (*anonymisation*),
- cascading revocation* (in which data or permissions revoked from one data controller are automatically revoked also from all parties with whom that controller has shared),
- consentless revocation* (this is the case in which one revokes data that has been used without explicit consent),
- delegated revocation* (in this case an individual confers upon another the ability to revoke).

The revocation types with an asterisk are derivative, while the others are basic; for instance, revocation of permissions to process data may be delegated and consentless. Cascading revocation is an ideal that is difficult to implement in practice; indeed, one goal of the EnCoRe project is to develop an implementation of this ideal which can be deployed in actual enterprise information systems.

III. Defining a Hoare Logic for Consent and Revocation

In this section we define a Hoare logic for consent and revocation processes, with a richer set of rights for principals.

Right	Meaning
$aO\delta$	a owns (originates) δ
$aL\delta$	a knows (where to locate) δ
$aP\delta$	a may process (personally identifiable) δ
$aA\delta$	a may aggregate (anonymous) δ
$aS\delta$	a may share δ (one-step further)
$aS^*\delta$	a may share δ transitively

Figure 1. Rights in the Hoare logic.

$$\begin{aligned} \psi &::= aO\delta \mid aL\delta \mid aP\delta \mid aA\delta \mid aS\delta \mid aS^*\delta \\ \Psi &::= \psi \mid \neg\Psi \mid \Psi_1 \wedge \Psi_2 \mid \Psi_1 \vee \Psi_2 \end{aligned}$$

Figure 2. Syntax of permissions in the Hoare logic.

We will later link the semantics of this logic to the access control model of Section V.

We now identify *six* distinct rights for principals, as explained in Figure 1. Most are self-explanatory; we have identified a special aggregate right aAx to be granted to include your data in anonymous aggregate data sets for purposes such as research.

The full syntax of permissions is shown in Figure 2; permissions are now defined by formulas of the form Ψ.

Next we identify the terms of the logic, and how these terms affect permissions and create obligations. The terms are explained in Figure 3 while the formal syntax of terms t and obligations Ω is given in Figure 4.

An obligation is a requirement on the state which results from applying one of the terms in the logic. While the effect of some terms is only to alter permissions, some terms result in requirements to apply further terms. We use the notation

$$\langle cond_1 \rangle\ t\ \langle cond_2 \rangle$$

to express obligations, with the following intuitive meaning: from a state satisfying $cond_1$ there is a requirement to apply term t to produce a new state satisfying $cond_2$.

The rules for the logic will be given in the form of Hoare triples, as follows:

$$\{\text{precondition}\}\, t\, \{\text{postcondition}\}$$

where pre- and postconditions are either permissions, obligations, or combinations of both.

We have three rules which describe in detail the effect of granting consent.

A principal a may grant consent for processing of a datum x to a principal b only if a owns δ or is able to share it. Once consent has been granted, b will know where to find δ and to process δ. Thus, the first rule for consent is as follows.

$\mathbf{grant}(a, b, \delta)$	grant consent for b to process δ
$\mathbf{grant}^1(a, b, \delta)$	grant consent δ for b to share onward once
$\mathbf{grant}^\dagger(a, b, \delta)$	grant consent δ for b to share onward transitively
$\mathbf{release}(a, b, \delta)$	release δ for anonymous aggregation at b
$\mathbf{revoke}(a, b, \delta)$	revoke permission for b to process δ (personally identifiably)
$\mathbf{revoke}^\dagger(a, b, \delta)$	cascade revoke permission for b and friends to process δ (personally identifiably)
$\mathbf{delete}(a, b, \delta)$	delete δ at b
$\mathbf{delete}^\dagger(a, b, \delta)$	cascade delete δ at b

Figure 3. Meaning of terms in the Hoare logic.

$$\begin{aligned} t &::= \mathbf{grant}(a, b, \delta) \mid \mathbf{grant}^1(a, b, \delta) \mid \\ &\quad\ \mathbf{grant}^\dagger(a, b, \delta) \mid \mathbf{release}(a, b, \delta) \mid \\ &\quad\ \mathbf{revoke}(a, b, \delta) \mid \mathbf{revoke}^\dagger(a, b, \delta) \mid \\ &\quad\ \mathbf{delete}(a, b, \delta) \mid \mathbf{delete}^\dagger(a, b, \delta) \\ qt &::= t \mid \forall c.t \text{ (where } c \text{ is bound in } t) \\ \Omega &::= \langle \Psi_1 \rangle\ qt\ \langle \Psi_2 \rangle \mid \Omega_1 \vee \Omega_2 \\ cond &::= \Psi \mid \Omega \mid \Psi \wedge \Omega \end{aligned}$$

Figure 4. Syntax of terms and obligations.

$$\begin{gathered} \{aO\delta \vee aS\delta\} \\ \mathbf{grant}(a, b, \delta) \\ \{bL\delta \wedge bP\delta\} \end{gathered}$$

A principal a may wish to allow another, b, not only to process some data δ, but also to share that data with principals to whom b is directly connected. This models the fact that in the real world, an enterprise may disclose data to a third party while restricting that party so that the data can only be shared under certain conditions, e.g. only to enterprises linked to the third party which are approved. For this type of consent, the granting principal a must own the data or have the right to share it; when consent is granted to principal b, b will be able to find, process and share δ by one step. This is all expressed by the rule

$$\{aO\delta \vee aS^*\delta\}$$
$$\mathbf{grant}^1(a, b, \delta)$$
$$\{bL\delta \wedge bP\delta \wedge bS\delta\}$$

In the case in which a allows b to share data onward, we have a generalisation of the term $\mathbf{grant}^1$, namely, the term $\mathbf{grant}^\dagger$. This enables b to share δ more than once, to all of its connections.

$$\{aO\delta \vee aS^*\delta\}$$
$$\mathbf{grant}^\dagger(a, b, \delta)$$
$$\{bL\delta \wedge bP\delta \wedge bS^*\delta \wedge bA\delta\}$$

To allow a datum δ to be aggregated, the granting principal a must own or have the right to share δ. This endows a principal b with knowledge of δ and the aggregation right $bA\delta$:

$$\{aO\delta \vee aS\delta\}$$
$$\mathbf{release}(a, b, \delta)$$
$$\{bL\delta \wedge bA\delta\}$$

Revocation of a datum δ can only be performed if a principal owns δ or has the right to share it; we are referring of course to revocation of permissions, which removes the ability of a principal b to process and share δ:

$$\{aO\delta \vee aS\delta\}$$
$$\mathbf{revoke}(a, b, \delta)$$
$$\{\neg bP\delta \wedge \neg bS\delta\}$$

Cascading revocation is more complex, as it creates obligations. A principal a performing cascading revocation from a principal b requires not only b to lose the privilege of processing δ; it obliges b to revoke δ from all other principals - forcing them to lose the ability to process - or simply to delete δ from all other principals. This is expressed using the following rule:

$$\{aO\delta \vee aS^*\delta\}$$
$$\mathbf{revoke}^\dagger(a, b, \delta)$$
$$\{\neg bP\delta \wedge (\langle bS^*\delta\rangle\ \forall c.\mathbf{revoke}^\dagger(b, c, \delta)\ \langle\neg bS^*\delta\rangle$$
$$\vee \langle\neg bS^*\delta \wedge bS\delta\rangle\ \forall c.\mathbf{delete}(b, c, \delta)\ \langle\neg bS\delta\rangle)\}$$

Deletion is expressed in the logic as a loss of knowledge and sharing rights. A principal a can delete a datum δ from another principal b if he or she owns or is able to share δ. The effect of deletion is to eliminate b's knowledge of x and b's ability to share δ:

$$\{aO\delta \vee aS\delta\}$$
$$\mathbf{delete}(a, b, \delta)$$
$$\{\neg bL\delta \wedge \neg bS\delta\}$$

Deletion can be cascaded, and this creates the obligation to prevent sharing of δ in all forms as well as knowledge of its existence, as shown in the rule:

$$\{aO\delta \vee aS^*\delta\}$$
$$\mathbf{delete}^\dagger(a, b, \delta)$$
$$\{\neg bL\delta \wedge \langle bS^*\delta\rangle\ \forall c.\mathbf{delete}^\dagger(b, c, \delta)\ \langle\neg bS^*\delta\rangle$$
$$\vee \langle\neg bS^*\delta \wedge bS\delta\rangle\ \forall c.\mathbf{delete}(b, c, \delta)\ \langle\neg bS\delta\rangle)\}$$

The above rules define a Hoare logic of consent and revocation. The rules allow us to reason about such processes, and they concisely express requirements for privacy and identity management systems which provide users with controls over personal data.

IV. Resolving Ambiguities and Limitations in the Logic

...

V. Defining an Access Control Model for Consent and Revocation Processes

Next we formalise the semantics of consent and revocation processes using labelled transition systems. What we propose in this section is effectively an *access control model* which specifically enables the requirements of such processes to be expressed.

We fix a set of principals $P = \{A, B, \ldots\}$, and a set of data objects $D = \{\delta_1, \delta_2, \ldots\}$. Principals are able to set permissions on objects, and grant and revoke consent. A *base permission* is defined as a subset of $\Sigma = \mathbf{2}^{\{c,p,d,c^\dagger,p^\dagger,d^\dagger\}} = \wp(\{c, p, d, c^\dagger, p^\dagger, d^\dagger\})$, where c, p, d denote respectively data collection, processing and disclosure rights. A *sharing right* is one of $c^\dagger, p^\dagger, d^\dagger$. In what follows the term *permission* will refer to a string of base or sharing rights.

A sharing permission represents the ability of a principal to endow another principal with the same permission as the former (this is known also as *delegation*). The right d is special: its presence indicates the ability of a principal to share the datum with another but not onward, while $d^\dagger$ enables a principal to share with another while enabling the other to share further.

DEFINE P HERE

We can define a rights matrix as a function

$$\rho : P \times D \to \Sigma \cup \{\mathsf{owner}, \mathsf{null}\} \tag{2}$$

The special rights owner and null are that of the originator of a datum, and the right to do nothing, respectively.The rights

matrix represents the permissions associated with the data, and may be easily presented in tabular form, as illustrated in the example below.

Example. Suppose we have a system of three principals A, B, C, who are manipulating two data, δ_1, δ_2. Principal B is the owner, or originator, of datum δ_2, and similarly principal C of δ_1. The entries in the matrix correspond to the value of ρ.

	δ_1	δ_2
A	cp	cpd
B	$cpd^\dagger$	owner
C	owner	p

In our model, the rights matrix ρ is part of the overall *consent and revocation state* ν of a given system. The consent and revocation state ν also contains the store S of all the data held by the different principals because both deletion and anonymisation affect data, not just permissions. In addition to the permissions, we need to include in the state information related to constraints, including a counter for the time that data has been held, a measure of the quantity (volume) of data held and its sensitivity. We may define the consent and revocation state as the tuple

$$\nu = (P, D, \rho, S, \tau, \omega, \kappa) \tag{3}$$

where $\tau, \omega \in \mathbb{R}, \kappa \in \{\texttt{SENSITIVEDPA}, \texttt{NONSENSITIVE}\}$ are precisely the information related to constraints listed in order above. The store S is simply a mapping from D to data values.

A principal can perform an operation $\mathbf{grant}(\sigma, \delta, \Phi, q)$ to endow another principal q with a consent permission for datum δ. Φ is a propositional formula expressing constraints over consent variables, such as (1). The syntax of constraints is as follows (for the value of *sensitivity* we assume two predefined constants, where SENSITIVEDPA represents the types of personal data deemed sensitive by the UK Data Protection Act; more constants could be added to represent other legislation):

$$\begin{aligned}
\Phi &::= \phi \mid \neg\Phi \mid (\Phi_1 \wedge \Phi_2) \mid (\Phi_1 \vee \Phi_2) \\
\phi &::= \psi_t \mid \psi_v \mid \psi_s \mid \psi_\Pi \\
\psi_t &::= t < n \mid t = n \\
\psi_v &::= v < n \mid v = n \\
\psi_s &::= s = \textit{sensitivity} \\
\psi_\Pi &::= \Pi \in R
\end{aligned}$$

where $R \subseteq P$ and n is an integer. In time constraints ψ_t, we intuitively regard n as a measure of time in calendar days (or seconds or years), while in volume constraints ψ_v is some measure of volume (at this time, we feel that volume may be useful variable, but we have not gathered enough evidence to decide what an appropriate measure would be; it is unlikely to be simply a value in kilobytes, it could even be some proportion of salient facts.)

Revocation of permission to process or to disclose are both captured by an action of the form $\mathbf{revokeperms}(\sigma, \delta, q)$, where σ is a permission, δ the datum and q the principal to whom the revocation is addressed; while $\mathbf{delete}(\delta)$ and $\mathbf{anonymise}(\delta)$ are the actions corresponding to the other two basic revocation types.

Consent and revocation processes are thus modelled by the actions $\mathbf{grant}(\sigma, \delta, \Phi, q)$, $\mathbf{revokeperms}(\sigma, \delta, q)$, $\mathbf{delete}(\delta)$, and $\mathbf{anonymise}(\delta)$, as interpreted over consent and revocation states: this is our basic model. Armed with the notations of this section, we can formalise the semantics of such processes.

The semantics of a consent and revocation process is a labelled transition system whose transitions are of the form

$$(r, action, \nu) \to \nu' \tag{4}$$

where $action ::= \mathbf{grant}(\sigma, \delta, \Phi, q) \mid \mathbf{delete}(\delta) \mid \mathbf{revokeperms}(\sigma, \delta, q) \mid$ and $q, r \in P$.

The tuple $(r, action, \nu)$ represents the action $action$ performed by principal $r \in P$ in the consent and revocation state ν. When r performs this action, the overall consent and revocation state changes to ν'. This is, effectively, a big-step operational semantics for consent and revocation processes.

The operation $\mathbf{grant}(\sigma, \delta, \Phi, q)$ simply updates the rights matrix with a new permission on datum δ for a principal $q \in P$ and ensures the resulting consent and revocation state satisfies Φ (the satisfaction relation $\models$ is defined later):

$$(r, \mathbf{grant}(\sigma, \delta, \Phi, q), \nu) \to \nu'$$
$$\text{where } \nu' = \nu[\rho \mapsto \rho'] \text{ such that } \nu' \models \Phi \text{ (otherwise } \nu' \text{ is undefined)}$$
$$\text{and } \rho'(x, y) = \begin{cases} \sigma \text{ for } x = q, y = \delta \\ \rho(x, y) \text{ otherwise} \end{cases}$$

The operation $\mathbf{delete}(\delta)$ removes the datum δ from the set D and ensures no value is associated with this symbol for any principal; it also eliminates permissions for this symbol by assigning the null permission null:

$$(r, \mathbf{delete}(\delta), \nu) \to \nu' \text{ where } \nu' = \nu[D \mapsto D', S \mapsto S', \rho \mapsto \rho']$$
$$\text{such that } D' = D - \{\delta\}, S'(x) = \begin{cases} \text{undefined if } x = \delta \\ S(x) \text{ otherwise} \end{cases}$$
$$\rho'(x, y) = \begin{cases} \text{null for } y = \delta \text{ for all } x \\ \rho(x, y) \text{ otherwise} \end{cases}$$

The operation $\mathbf{revokeperms}(\sigma, \delta, q)$ reduces the permissions of principal q by σ. This can only be performed by the

owner of δ.

$$(r, \mathbf{revokeperms}(\sigma, \delta, q), \nu) \rightarrow \nu' \text{ where } \nu' = \nu[\rho \mapsto \rho']$$
$$\rho'(x,y) = \begin{cases} \rho(x,y) - \sigma \text{ for } x = q, y = \delta \text{ if } \rho(r,\delta) = \mathsf{owner} \\ \rho(x,y) \text{ otherwise} \end{cases}$$

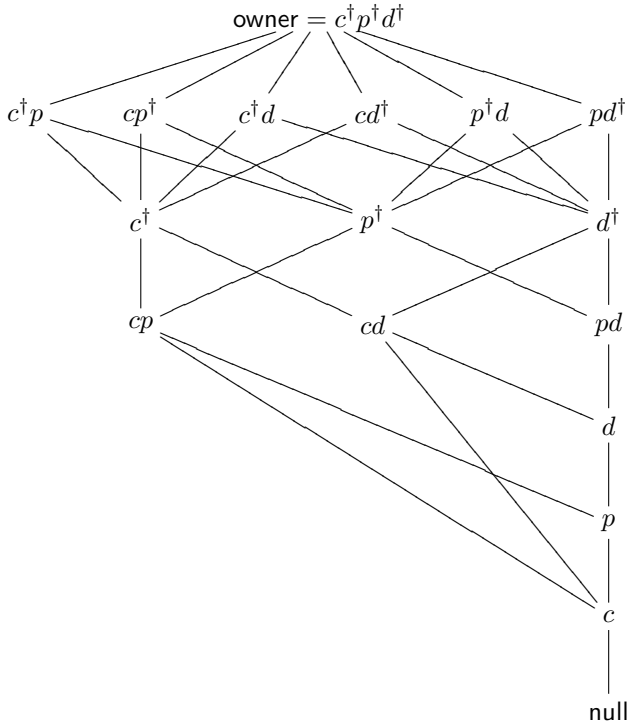
The operation $\mathbf{anonymise}(\delta)$ is defined below as associating all principals in P who currently have permission to process or disclose δ as now having maximal permissions over a suitably modified $\overline{\delta}$ representing an appropriately anonymised version $\overline{\delta}$ of the data δ:

$$(r, \mathbf{anonymise}(\delta), \nu) \rightarrow \nu' \text{ where } \nu' = \nu[\rho \mapsto \rho']$$
$$\text{such that } \forall x \in Q \ : \rho'(x, \overline{\delta}) = \mathsf{owner}$$
$$\text{where } Q = \{x \in P : p \in \rho(x,\delta) \wedge d \in \rho(x,\delta)\}$$

Note that the set Q consists of all principals x with processing (p) and dissemination (d) rights on δ. We do not underestimate the challenge in producing an appropriately modified $\overline{\delta}$! Within the EnCoRe research programme, we will be considering how to ensure that such $\overline{\delta}$ cannot be aggregated to compromise privacy. We will also be addressing how one might check claims made about algorithms for generating $\overline{\delta}$.

Note that we have not affected rights to the original δ. If it is desired to delete the data or revoke permissions to it, that will need to be done separately.

The permissions in our model form a hierarchy, namely a lattice with upper bound the maximal permission owner, which we define as an abbreviation for $c^\dagger p^\dagger d^\dagger$, and lower bound the special value null. We intend to develop theoretical results about this structure in an analogous manner to [22].



Satisfaction Relation for Constraints.: We interpret constraints Φ over consent and revocation states ν using a satisfaction relation $\models$, defined as follows.

$$\begin{aligned}
\nu \models \psi_t \quad &\text{if} \quad \nu \models_t \psi_t \\
\nu \models \psi_v \quad &\text{if} \quad \nu \models_v \psi_v \\
\nu \models \psi_s \quad &\text{if} \quad \nu \models_s \psi_s \\
\nu \models \psi_\Pi \quad &\text{if} \quad \nu \models_\Pi \psi_\Pi \\
\nu \models \psi_{act} \quad &\text{if} \quad \nu \models_{act} \psi_{act} \\
\nu \models \neg\phi \quad &\text{if} \quad \nu \not\models \phi \\
\nu \models (\phi_1 \wedge \phi_2) \quad &\text{if} \quad \nu \models \phi_1 \text{ and } \nu \models \phi_2 \\
\nu \models (\phi_1 \vee \phi_2) \quad &\text{if} \quad \nu \models \phi_1 \text{ or } \nu \models \phi_2
\end{aligned}$$

With $\nu = (P, D, \rho, S, \tau, \omega, \kappa)$ we define the satisfaction relations $\models_t, \models_v, \models_s, \models_\Pi, \models_a$ as follows:

$$\nu \models_t t < n \text{ if } t < \tau, \quad \nu \models_t t = n \text{ if } t = \tau$$
$$\nu \models_v v < n \text{ if } v < \omega, \quad \nu \models_v v = n \text{ if } v = \omega$$
$$\nu \models_s s = \textit{sensitivity} \text{ if } \kappa = \textit{sensitivity}$$
$$\nu \models_\Pi \Pi \in R \text{ if } \forall r \in R, d \in S, \rho(r,\delta) = \mathsf{owner} \text{ or } \rho(r,\delta) \geq p$$

Note that $\rho(r,\delta) \geq p$ states that the permission of principal r on item d should contain p, i.e. the right to process d.

Revocation Criteria.: We can formally define criteria for correct revocation, deletion and anonymisation by creating a relation $\vdash$ between states ν and actions.

$$\begin{aligned}
\nu \vdash \mathbf{delete}(\delta) \quad &\text{if} \quad \delta \notin D \text{ and } S(\delta) = \text{undefined} \\
\nu \vdash \mathbf{revokeperms}(\sigma, \delta) \quad &\text{if} \quad \delta \in D, \delta \in S, \text{ and for all } r, \\
&\qquad \rho(r,\delta) = (\mathsf{owner} - \sigma) \\
\nu \vdash \mathbf{anonymise}(\delta) \quad &\text{if} \quad \text{all principals associated with } \delta \\
&\qquad \text{are now also associated with } \overline{\delta}, \\
&\qquad \forall r \in P : \rho(r,\delta) = \mathsf{owner}
\end{aligned}$$

The semantic model of consent and revocation processes proposed up to this point is simple and incorporates some of the elements of interest for analysis of systems. We have noted that anonymisation poses some interesting issues with regard to ownership and rights over data. In the next section we consider a richer model of consent and revocation processes, presented in the style of a Hoare logic.

VI. Linking the Two Models

...

VII. Applications

In this section we describe how the access control model and logic might be applied to practical scenarios.

Reasoning about privacy policies.: Privacy policies typically express permissions and obligations of data controllers. The best means of expressing rights and obligations is hotly debated, and some relevant investigations have been cited already in the Related Work section.

With the models proposed so far in this paper, we have formal languages for reasoning about rights and obligations. In particular, our rules express how rights and obligations are affected and created through the expression of a user's privacy preferences, namely, through the actions of consent and revocation.

Consider P3P policies — these are XML files consisting of policy *statements*. A statement is a policy rule expressing how a data value can be manipulated. Statements have four fields:

- `<DATA-GROUP>`, which identifies the type of data value to which the statement applies,
- `<PURPOSE>`, which takes one of many predefined values in the P3P specification [15],
- `<RECIPIENT>`, which specifies to whom the data value may be provided, and
- `<RETENTION>`, which is a constraint on how long data may be held for by the data controller.

The access control model presented in Section V already accounts for the elements of a P3P policy statement: the purpose, intended recipients and retention constraints for a datum can be expressed by writing suitable formulae involving consent variables.

A policy is of course a static definition of rights, and using our access control model we can consider the dynamics of rights as users grant and revoke consent. A policy may specify that data is to be deleted, but the act of deletion itself needs to be performed by a piece of software (typically) known as a *policy enforcement point*, or PEP (see the literature on Privacy RBAC [11], for example, for information on policy architectures). The rules for the operations in our access control model provide specifications of how such enforcement points should operate, namely, what the consent and revocation state should be after a user operation.

There is one limitation to the access control model we have proposed, which is remedied in the Hoare logic: the first model deals only implicitly with obligations, and only with one type of obligation related to persistence of data. An obligation is expressed using a predefined string, which in example (1) is `delete`, but obligations are more complex, justifying the richer syntax used in our logic.

The logic for consent and revocation provides a more direct means of reasoning about rights and obligations in particular. The rules in a privacy policy could be expressed as a set of permissions on data; this transformation might be automated. The logic then provides a way of understanding how permissions change, and in particular one might construct proofs that a particular data controller does or does not have particular permissions.

Expressivity of the two formalisms.: The access control model and the logic have different expressive powers. It is easy, for instance to construe sharing permissions: one might be tempted to treat the access control permissions d and $d^\dagger$ for some principal a on a datum x as roughly equivalent to the logic permissions aSx and aS^*x. However, the intention is completely different.

The logic permissions allow a principal to control with how many others a datum can be shared; aSx allows sharing with only principals that are directly linked to a (note that information on who is linked to whom must be assumed), while aS^*x allows cascading sharing, i.e. for x to be shared with all principals that are linked to all principals linked with a.

On the other hand, the access control permissions express the ability of one principal to endow another with a particular right. While the permission d allows a principal to share (disseminate) a datum, the permission $d^\dagger$ allows that principal to endow another with the ability to share. This is a subtle but important distinction, and we expect it will make a difference in specifications of large scale systems which include privacy controls, such as online social networks.

A Simple Example.: Using the rules of section III, one can prove which permissions hold after a specified sequence of actions. Through a simplistic example we will be able to illustrate some of the issues and ambiguities which our models are designed to resolve.

Consider a fictitious scenario in which there are two principals, a, b and three data values x, y, z such that a owns x and y, while b owns z. This state may be represented as a consent and revocation state ν_1 with $P = \{a, b\}$, $D = x, y, z$,

$$S(d) = \begin{cases} \text{“Jo Corsini” for } d = x \\ \text{“49 West Blvd.” for } d = y \\ \text{“New Crompton” for } d = z \end{cases}$$

and a rights matrix ρ as follows:

	x	y	z
a	owner	owner	
b			owner

We can express the rights of the principals in the logic using the permission

$$aOx \wedge aOy \wedge bOz$$

Suppose that a grants to b the ability to *process and share* y. This statement may at first seem ambiguous: does the previous sentence refer to the permission $p^\dagger$ or to the permission pd ? In fact, it refers to the latter, since the right $p^\dagger$ would enable b not only to process y, but to allow others to process y. Therefore the action being performed is

$$(a, \mathbf{grant}(pd, y, b), \nu_1) \to \nu_1'$$

where $\nu_1' = \nu_1[\rho \mapsto \rho']$ and ρ' is

	x	y	z
a	owner	owner	
b		pd	owner

In the logic, the action just performed is described succinctly by the term $\mathbf{grant^1}(a, b, x)$. According to the rule for $\mathbf{grant^1}$, this term can be applied if a is the owner of x (which is true in this example), and the effect is to create the permission $bLx \wedge bPx \wedge bSx$.

VIII. Conclusions and Future Work

The contributions of this paper are twofold: we have defined two formal models of the processes of consent and revocation, an access control model and a Hoare logic. Both models provide us with ways of reasoning about the state of personal data and the corresponding rights or permissions. They allow us to prove correctness properties of systems providing privacy controls expressed in the form of consent and revocation preferences for the handling of personal data.

The logic we have proposed exhibits most of the properties of consent and revocation, and we aim to validate our choice of terms through field research and applications to large scale examples, including a system design for managing data held in biobanks and other EnCoRe project case studies.

We are currently studying the lattice structure of consent and revocation permissions; a careful analysis of this structure will allow us to formally define more operations on permissions, thus resulting in richer models. In particular, we have in this paper assumed that the granting of a permission replaces any previous permission that a principal may have for a particular datum; however, consecutive $\mathbf{grant}$ operations in a real system are likely to produce cumulative permissions rather than replacing the current permission completely — for this to be modelled we need a suitable notion of addition and an well-defined ordering of permissions. Equally, to implement consecutive revocation actions a permissions stack may be needed; this would provide a history of permissions generated by consent and revocation actions, and revocation would be treated as a step backward into a previously granted permission. This is an area for future investigations.

The models we have proposed pave the way for implementation of tools for managing personal data online, and we intend to use them as the basis for developing a formal verification framework specifically for privacy management systems. Early work on formal verification of P3P policies using the CSP/FDR toolset have been presented in [19]. We aim to build a full verification tool implementing the logic for consent and revocation processes. Our hope is that this will enable reasoning not only about specific privacy policies, but about the capabilities of individuals and data controllers in sensitive data sharing scenarios.

References

[1] BBC News, "Data loss firm contract axed," 10 September 2008, http://news.bbc.co.uk/1/hi/uk_politics/7608155.stm.

[2] S. Creese and K. Lamberts, "Can cognitive science help us make information risk more tangible online?" *IEEE Intelligent Systems*, vol. 24, no. 6, pp. 32–36, 2009.

[3] D. Solove, "Identity theft, privacy, and the architecture of vulnerability," *Hastings Law Journal*, vol. 54, p. 1227, 2003.

[4] D. J. Solove, "A taxonomy of privacy," *University of Pennsylvania Law Review*, vol. 154, no. 3, pp. 477–560, January 2006. [Online]. Available: #

[5] J. Plamenatz, *Consent, Freedom and Political Obligation*. OUP, 1963.

[6] R. F. T.L. Beauchamp, *A History and Theory of Informed Consent*. OUP, 1986.

[7] I. Agrafiotis, S. Creese, M. Goldsmith, and N. Papanikolaou, "Reaching for informed revocation: Shutting off the tap on personal data," in *Proceedings of Fifth International Summer School on Privacy and Identity Management for Life*, September 2009.

[8] M. Burrows and M. Abadi, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18—36, February 1990.

[9] M. Abadi and M. R. Tuttle, "A semantics for a logic of authentication," in *Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing*. ACM, August 1991, pp. 201—216.

[10] M. Abadi, M. Burrows, and B. Lampson, "A calculus for access control in distributed systems," *ACM Transactions on Programming Languages and Systems*, vol. 15, no. 4, pp. 706—734, September 1993.

[11] Q. Ni, A. Trombetta, E. Bertino, and J. Lobo, "Privacy-aware role based access control," in *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*. New York, NY, USA: ACM Press, 2007, pp. 41–50.

[12] M. Y. Becker, A. Malkis, and L. Bussard, "A framework for privacy preferences and data-handling policies," Microsoft Research, Technical Report MSR-TR-2009-128, 2009.

[13] M. J. May, C. A. Gunter, I. Lee, and S. Zdancewic, "Strong and weak policy relations," University of Pennsylvania, Technical Report MS-CIS-09-10, 2009.

[14] C. A. Gunter, M. J. May, and S. G. Stubblebine, "A formal privacy system and its application to location based services," in *Proceedings of PET 2004*, ser. Lecture Notes in Computer Science, D. Martin and A. Serjantov, Eds., vol. 3424, 2005, pp. 256—282.

[15] L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle, "The platform for privacy preferences 1.0 (p3p1.0) specification," W3C Recommendation, Apr. 2002. [Online]. Available: http://www.w3.org/TR/P3P/

[16] G. Hogben, "A technical analysis of problems with p3p v1.0 and possible solutions," November 2002, position paper for W3C Workshop on the Future of P3P. Available at http://www.w3.org/2002/p3p-ws/pp/jrc.html.

[17] R. Wenning, "Private communication," November 2009.

[18] M. Casassa Mont, S. Pearson, G. Kounga, Y. Shen, and P. Bramhall, "On the management of consent and revocation in enterprises: Setting the context," HP Laboratories, Technical Report HPL-2009-49, 2009.

[19] N. Papanikolaou, S. Creese, and M. Goldsmith, "Policy refinement checking," in *Proceedings of Ninth International Workshop on Automated Verification of Critical Systems (AVoCS 09)*, L. O'Reilly and M. Roggenbach, Eds., September 2009.

[20] M. Casassa Mont, S. Pearson, S. Creese, M. Goldsmith, and N. Papanikolaou, "Towards an integrated approach to the management, specification and enforcement of privacy policies," in *Proceedings of W3C Workshop on Access Control Application Scenarios*, November 2009.

[21] S. Patil, "Why is evaluating usability of privacy designs so hard? lessons learned from a user study of prism," in *Proceedings of iSociety Conference 2009*, 2009.

[22] D. E. Denning, "A lattice model of secure information flow," *Communications of the ACM*, vol. 19, no. 5, pp. 236—243, May 1976.