Privacy and Contextual Integrity: Framework and Applications

Adam Barth Anupam Datta John C. Mitchell
Stanford University
{abarth, danupam, jcm}@cs.stanford.edu

Helen Nissenbaum
New York University
helen.nissenbaum@nyu.edu

Abstract

Contextual integrity is a conceptual framework for understanding privacy expectations and their implications developed in the literature on law, public policy, and political philosophy. We formalize some aspects of contextual integrity in a logical framework for expressing and reasoning about norms of transmission of personal information. In comparison with access control and privacy policy frameworks such as RBAC, EPAL, and P3P, these norms focus on who personal information is about, how it is transmitted, and past and future actions by both the subject and the users of the information. Norms can be positive or negative depending on whether they refer to actions that are allowed or disallowed. Our model is expressive enough to capture naturally many notions of privacy found in legislation, including those found in HIPAA, COPPA, and GLBA. A number of important problems regarding compliance with privacy norms, future requirements associated with specific actions, and relations between policies and legal standards reduce to standard decision procedures for temporal logic.

1 Introduction

In the past few decades, we have seen a radical intensification in the social practices of gathering, storing, manipulating, and sharing information about people (henceforth, “personal information”). In many instances, new practices have aroused suspicion, indignation, and protest not only among legal experts, social critics, and privacy advocates, but also in the popular media and among the general public. Recent controversies range from the introduction of Caller ID to Lotus Marketplace Households and EZ Pass, from Carnivore and “total information awareness” to Internet cookies and online profiling. While there are philosophical theories of the nature and value of privacy, these tend to offer an account of what privacy is—say, control over information about oneself—and may explain why it ought to be valued and protected in liberal democracies. In contrast, the framework of *contextual integrity* has arisen in recent years to provide guidance on how to respond to conflicts between values and interests and to provide a systematic setting for understanding privacy expectations and the reasons that certain events cause moral indignation [30, 32].

This paper presents a formal framework for expressing privacy expectations and privacy practices, inspired by contextual integrity. We begin with a simple model of the transmission of personal information, containing communications such as “Alice gives Bob a certain type of information about Charlie,” and use first-order temporal logic for expressing and reasoning about norms of transmission. The central concepts drawn from contextual integrity include contexts, roles, and a focus on the type of information transmitted (Charlie’s height) rather than specifics of the data (Charlie is 5’10” tall). Roles within contexts are used to express that communication which is perfectly acceptable between a psychiatrist and patient is completely unacceptable between a human resource specialist and a job applicant. Temporal logic with *past* and *future* operators is used to say, for example, that certain information may be disclosed only if the subject mentioned has previously given permission or that if certain information is made public, notification must be sent to the concerned party. While contextual integrity was developed to support specific, substantive philosophical and legal positions, our goal is to formalize concepts from contextual integrity so that privacy guidelines, policies, and expectations can be stated precisely, compared, and enforced by an information processing system.

We define two kinds of norms, which we call *positive* and *negative*, as temporal logic formulas of two certain forms. These two kinds of norms generalize “allow” and “deny” rules in traditional access control for our setting with temporal conditions. A positive norm permits communication *if* its temporal condition is satisfied, whereas a negative norm permits communication *only if* its temporal condition is satisfied. These norms are interpreted in a model of communicating agents who “respect” the norms if the trace history of their communication satisfies a temporal formula constructed from the norms by taking the disjunction over positive norms and the conjunction over negative norms.

A communication action transmits information about a subject from a sender to a recipient. Our model of "information" includes a relation enabling agents to combine messages to compute additional information about the subject (e.g., computing postal code from postal address), elucidating the notion of a "data hierarchy" found in P3P [15] and EPAL [27]. To illustrate the expressiveness of this framework and explain its use, we show how to capture privacy provisions of HIPAA, COPPA, and GLBA as combinations of positive and negative norms with temporal conditions.

A number of frameworks for defining and enforcing access control and privacy policies have been proposed, including RBAC [12, 14, 25], EPAL [7, 8, 38], and P3P [1, 2, 13, 15, 16, 34]. In comparison with access control and previous privacy policy frameworks, our norms focus on who personal information is about, how it is transmitted, and past and future actions by both the subject and the users of the information. Generally speaking, access control policies enable a system to decide whether to allow or deny a specific action, typically by deriving a relation between subjects, objects, and actions (possibly by grouping subjects by role, etc.). Conventional access control systems might make decisions based on the current state of the resources that it governs, but generally do not inquire about the past or impose restrictions on the future. Some privacy policy languages, such as EPAL, have a rudimentary temporal nature, in that a request to perform an action might lead to an to allow or deny judgment and an "obligation." In EPAL, an obligation is usually an action that some agent is required to perform in the future. Our norms can refer explicitly to past and future actions using temporal operators. Thus, the decision to allow an action can depend on what actions have occurred previously and can require additional actions in the future, capturing both "opt-in" (a past requirement) and confidentiality (a future requirement) using a single construct.

Access control does not conventionally track whom information is about: permission to read or write a file might be granted or denied, but the decision is not based on who is described by the information in the file. In our model, the subject of information in a message is as important as the sender and the recipient of the message. For example, norms can permit doctors to communicate personal information about their patients but forbid them from communicating the personal information of their administrative assistants.

Like much of the work on access control and privacy languages in the computer security community, we express privacy policies in a formal logic and relate issues of compliance and refinement to the logical concepts of satisfiability and entailment. Specific technical results in Sect. 4 include characterizations of policy consistency, entailment, and compliance in Linear Temporal Logic (e.g., [29]). Entailment is key to understanding how to combine policies, and how to compare one policy, such as HIPAA, with another, such as the specific privacy practices of a clinic and hospital. Previous work on privacy languages, particularly EPAL, used a complex lattice-based definition of entailment. In our model, entailment is captured as standard logical implication. Policy combination is then achieved through the usual logical operations of conjunction and disjunction.

Our current framework makes two simplifications: norms are based only on the type of information communicated and information is assumed to describe an individual rather than a group of individuals. For example, we can easily express that it is acceptable for a physician to record certain types of information, but it is outside the scope of our current language to say that the average salary of bank managers can be released only if it does not identify a particular individual's salary. We believe it will be fruitful to develop precise connections with research on data privacy and aggregation in the future, but for simplicity we do not consider these extensions in the current paper.

The remainder of the paper is organized as follows. Section 2 overviews contextual integrity. Section 3 contains our model and defines our formal language CI. Section 4 investigates properties of and relations between policies expressed CI. Section 5 evaluates the expressiveness of CI by encoding privacy provisions from legislation in the language. Section 6 compares our framework with several access control and privacy languages. Section 7 concludes.

2 Overview of Contextual Integrity

Contextual integrity is a philosophical account of privacy in terms of the transfer of personal information. It is not proposed as a full definition of privacy, but as a normative model, or framework, for evaluating the flow of information between agents (individuals and other entities), with a particular emphasis on explaining why certain patterns of flow provoke public outcry in the name of privacy (and why some do not). In the approach encompassed by contextual integrity, the intricate systems of social rules governing information flow are the crucial starting place for understanding normative commitments to privacy. While *contextual integrity* is itself a relatively recent term, the idea of contextually relative norms has been "in the air," recognized in various ways in the literature (e.g., [33, 36, 37]), and explored in some specific ways in a variety of work dealing with professional confidentiality rules. Four constructs are key to defining contextual integrity: informational norms, appropriateness, roles, and principles of transmission.

We begin, however, with concept of a *context* to capture the idea that people act and transact in society not simply as individuals in an undifferentiated social world, but as individuals in certain capacities (roles), in distinctive social contexts, such as health care, education, employment, the marketplace, and so on. These contexts should be understood as structured settings whose features have evolved over time—sometimes long periods of time—subject to a host of contingencies of place, culture, historical events, and more. Features which characterize particular contexts include the assemblages of roles (sometimes open-ended) and the set of behavior-guiding norms that prescribe (and proscribe) actions and practices, when, for example, people consult a physician (or are the physician), attend school (or teach), and shop (or sell).

One further feature is key to understanding what we mean here by "contexts," for not only are they characterized by roles and norms but also by certain ends, or values. In the case of health care, an onlooker (say, from another planet) observing a typical health care setting of a hospital, will be unable make proper sense of the goings-on without appreciating the underlying purpose behind it, that is, alleviating illness and promoting health. Although settling the exact nature of the ends and values for any given context is not a simple matter—even in the case of health care, which is relatively robust—the central point is that the roles and norms of a context make sense, largely, in relation to them. Because this point, though relevant to the larger theory of contextual integrity, is not crucial to the specific goals of this paper, we will not elaborate on it any further. Instead, our formalization deals with contexts frozen at a particular moment in history, focusing on expressing their attendant norms precisely.

For purposes of understanding privacy, norms that apply to the transmission (or communication) of personal information from one party to another, which we call "informational norms," are singularly important. In a health care context, for example, informational norms limit what physicians can say to others about the health condition of patients under their care. Contextual integrity, then, is a feature of situations in which the informational norms of a context have been respected; when any of these norms have been unjustly breached, than we say that contextual integrity has been violated.

One of the key defining aspects of informational norms, and judgments that contextual integrity has or has not been violated, is the type (category, nature, class) of information in question. Unlike a number of prominent normative accounts of privacy, the approach taken here rejects the idea that a simple dichotomy—usually between public and private (sensitive, intimate) information—is sufficient for adjudicating privacy claims. Instead, there is potentially an indefinite variety of types of information that could feature in the informational norms of a given context. We suggest the term "appropriateness" as a way to signal whether the type of information in question conforms to the relevant informational norms. Thus, for example, in the context of a job interview for the position of bank manager in the present-day United States, information about applicants' marital status is inappropriate, but it is appropriate in the context of dating (or courtship). Because information type is so salient an influence on people's judgments that a violation has occurred, earlier accounts of contextual integrity had posited norms of appropriateness as distinct from norms of transmission. Our effort to formalize contextual integrity has revealed, however, that, at a certain level of generality, both can be covered by the form of transmission norm explored in this paper.

Associated with every communication there are three relevant entities (agents, principals): the one from whom the information flows, the one to whom the information flows, and the one—the information subject—about whom the information is. Entities are considered to be acting in certain capacities, or roles, which are articulated with varying degrees of detail, within the relevant contexts. In academic departments, for example, the roles of chair, tenured faculty, assistant professor, student, administrator, and so forth, each are associated with a set of duties and privileges. Thus, contextual integrity maintains that roles are key variables affecting the rich and complex sensibility people demonstrate in their judgments over whether a violation has occurred.

The notion of a transmission principle may be the most distinctive aspect of the approach to privacy through contextual integrity. These principles are the specific constraints (terms or conditions) regulating flow of information from entity to entity prescribed by informational norms. One such principle is confidentiality, prohibiting agents receiving information from sharing it with others in the future. Although confidentiality is prominent, there are many other principles of transmission, for example, reciprocity, determining that information flow is bi-directional (occurring in friendship but not between a patient and a physician). Another is dessert, determining that an agent deserves to know or learn something about the subject, perhaps, people deserving to know whether their lovers are HIV positive. An important family of transmission principles hinges on the awareness and consent of the information subject; in one instance, a subject might be forced to reveal information, in another, a subject might know (or not know) whether information has been transmitted, in a third, the subject consents to transmit information, and so on. Norms prescribe which transmission principles ought to govern the flow of information and is understood to be violated if the principles are not followed. It is worth noting that control by subjects of the flow of information about themselves, which features definitively in certain theories, is merely one transmission principle—albeit an important one—among many. There is probably no end to the variation in transmission principles.

3 A Formal Model of Contextual Integrity

In this section, we formalize a fragment of contextual integrity. Our model consists of communicating agents who take on various roles in contexts and send each other messages containing attributes of other agents. The evolution of the knowledge of individual agents depends on messages they receive and computation rules that enable agents to infer further attributes. Agent interactions give rise to execution histories, or traces. In our specific model, norms of transmission are expressed using Linear Temporal Logic (LTL) formulas interpreted over these traces, although the choice of linear time over other temporal logics may not be highly significant.

3.1 Agents, Attributes, and Messages

We begin by modeling communicating agents. Associated with each agent is a collection of the attributes that agent knows. Let $\mathcal{P}$ be a set of *agents*, and let $\mathcal{T}$ be a set of *attributes*. For example, Alice and Bob are agents, and "postal address" and "height" are attributes. A *knowledge state* κ is a subset of $\mathcal{P} \times \mathcal{P} \times \mathcal{T}$. If $(p, q, t) \in \kappa$, we say agent p *knows* the value of attribute t of agent q. For example, Alice knows Bob's height. We omit "group" attributes, for example the average height of Alice, Bob, and Charlie.

Data Model. To structure attributes, we include computation rules. Our computation rules provide an abstract presentation of possible inferences, enabling agents to compute the attribute "postal code" from the attribute "postal address". Formally, a *computation rule* is a pair (T, t), where $T \subseteq \mathcal{T}$ and $t \in \mathcal{T}$. Intuitively, if Alice knows the value of each attribute in T for Bob, then Alice can compute the value of attribute t for Bob. We express this formally as a relation on knowledge states:

$$\forall \kappa . \forall p, q \in \mathcal{P}.\text{if } \{p\} \times \{q\} \times T \subseteq \kappa, \text{ then } \kappa \xrightarrow{(T,t)} \kappa'$$

where $\kappa' = \kappa \cup \{(p, q, t)\}$. That is, agent p learns attribute t about agent q. Let I be a set of computation rules. The relation $\xrightarrow{I}$ is the transitive closure of $\xrightarrow{(T,t)}$ for $(T, t) \in I$.

Communication Model. An agent can send a message to another agent provided the sending agent knows all the attributes communicated by the message. For example, Alice can send a message to Bob containing Charlie's height just in case Alice herself knows Charlie's height. After receiving such a message, Bob learns Charlie's height. Messages m are drawn from a set $\mathcal{M}$. Associated with each message m is a (possibly empty) set of attributes which the message contains, $\text{content}(m) \subseteq \mathcal{P} \times \mathcal{T}$, which is closed under computation rules. For example, a message that contains a postal address necessarily contains the corresponding postal code. We refer to the act of sending a message as a *communication action* and represent such actions as triples (p_1, p_2, m), where agent p_1 is the *sender*, agent p_2 is the *recipient*, and m is the message being sent. A communication action transforms knowledge states as follows:

$$\begin{aligned} &\forall \kappa, \hat{\kappa} . \forall p_1, p_2 \in \mathcal{P}. \forall m \in \mathcal{M}. \\ &\qquad \text{if } \kappa \xrightarrow{I} \hat{\kappa} \text{ and } \{p_1\} \times \text{content}(m) \subseteq \hat{\kappa}, \\ &\qquad\qquad \text{then } \kappa \xrightarrow{(p_1,p_2,m)} \kappa', \end{aligned}$$

where $\kappa' = \hat{\kappa} \cup \{p_2\} \times \text{content}(m)$. The contents of the message are first computed by the sender (at $\hat{\kappa}$) and then learned by the recipient (at κ').

3.2 Roles, Contexts, and Traces

In order to model contextual integrity, we impose additional structure that associates agents with roles as part of contexts. Let $\mathcal{R}$ be a set of *roles* and $\mathcal{C}$ be a partition of $\mathcal{R}$. We refer to elements $c \in \mathcal{C}$ as *contexts* and the roles $r \in c$ as the roles of context c. For example, "teller" is a role in a banking context and "doctor" is a role in a health care context. The roles are structured by a partial order $\leq_{\mathcal{R}}$. If $r_1 \leq_{\mathcal{R}} r_2$, then r_1 is a specialization of role r_2 and, symmetrically, r_2 is a generalization of r_1. For example, a psychiatrist is a specialization of a doctor, which in turn is a specialization of a health care provider.

Agents can be active in multiple roles simultaneously. For example, Alice can be at once a doctor in a health care context and a customer in a banking context. A *role state* ρ is a subset of $\mathcal{P} \times \mathcal{R}$. If $(p, r) \in \rho$, we say agent p is active in, or plays, role r. For example, if $(\text{Alice}, \textit{psychiatrist}) \in \rho$, then Alice is active in the role of psychiatrist. We require role states to be closed under role generalization, that is if $r_1 \leq_{\mathcal{R}} r_2$ and $(p, r_1) \in \rho$, then $(p, r_2) \in \rho$. Returning to our example, if $(\text{Alice}, \textit{psychiatrist}) \in \rho$, Alice must be active in the role of doctor in addition to that of psychiatrist. There are many instances of each context (many banks, many hospitals), but for clarity we omit instances.

The history of the agent world is an (infinite) *trace*: a sequence of triples (κ, ρ, a), where κ is a knowledge state, ρ is a role state, a is a communication action, and

$$\kappa_n \xrightarrow{a_{n+1}} \kappa_{n+1}, \text{ for all } n \in \mathbb{N}.$$

The role state can change freely from one state to the next. We view the role state as an input to the model. For example, a hospital provides as input to the policy mechanism a record of which of its employees are nurses, which are doctors, etc. The knowledge state, however, evolves in concert with the communication actions. This prevents Alice from spontaneously learning Charlie's birthday.

$$\sigma \models \Box \forall p_1, p_2, q : P. \forall m : M. \forall t : T.$$
$$\text{incontext}(p_1, c) \wedge \text{send}(p_1, p_2, m) \wedge \text{contains}(m, q, t) \rightarrow \bigvee_{\varphi^+ \in \text{norms}^+(c)} \varphi^+ \wedge \bigwedge_{\varphi^- \in \text{norms}^-(c)} \varphi^- \quad (1)$$

$$\begin{aligned} &\text{positive norm:} \quad \text{inrole}(p_1, \hat{r}_1) \wedge \text{inrole}(p_2, \hat{r}_2) \wedge \text{inrole}(q, \hat{r}) \wedge (t \in \hat{t}) \wedge \theta \wedge \psi \\ &\text{negative norm:} \quad \text{inrole}(p_1, \hat{r}_1) \wedge \text{inrole}(p_2, \hat{r}_2) \wedge \text{inrole}(q, \hat{r}) \wedge (t \in \hat{t}) \wedge \theta \rightarrow \psi \end{aligned}$$

Figure 1. Norms of Transmission Represented as a Temporal Formula

3.3 Temporal Logic

We employ a standard temporal logic for expressing properties of traces of agent actions (e.g., [29]). The temporal operators are used to capture the principles of transmission. For example, if Alice tells Bob her age under the principle of confidentiality, then, in the future, Bob must not disclose Alice's age. The past operators are also useful for capturing "opt-in" and other similar privacy idioms. Several temporal logics are appropriate for formalizing contextual integrity, including linear temporal logic and branching-time temporal logic. We employ linear temporal logic, in particular multi-sorted, first-order LTL. The interested reader can find the details in Appendix A. We use formulas generated by the following grammar:

$$\begin{aligned} \varphi ::= &\ \text{send}(p_1, p_2, m) \mid \text{contains}(m, q, t) \mid \\ &\ \text{inrole}(p, r) \mid \text{incontext}(p, c) \mid t \in t' \mid \\ &\ \varphi \wedge \varphi \mid \neg\varphi \mid \varphi \mathcal{U} \varphi \mid \varphi \mathcal{S} \varphi \mid \bigcirc \varphi \mid \exists x : \tau . \varphi \end{aligned}$$

Intuitively, $\text{send}(p_1, p_2, m)$ holds in a state if agent p_1 just sent message m to agent p_2, $\text{contains}(m, q, t)$ holds if message m contains attribute t of agent q, $\text{inrole}(p, r)$ holds in a state if agent p is active in role r, $\text{incontext}(p, c)$ holds in a state if agent p is active in a role of context c, $t \in t'$ holds if attribute t can be computed from (is a component of) attribute t', and $\varphi \mathcal{U} \psi$ holds just in case φ holds until ψ holds (ψ must eventually hold). The modality "since," written $\mathcal{S}$ is the past version of $\mathcal{U}$. $\bigcirc\varphi$ holds iff φ holds in the next state. Finally, $\exists$ is rigid existential quantification.

To simplify notation, we use the following standard symbols: $\Diamond$ for "eventually," $\Box$ for "henceforth," $\diamondminus$ and $\boxminus$ for the past versions of $\Diamond$ and $\Box$, respectively, and $\mathcal{W}$ for "wait for." The formula $\varphi \mathcal{W} \psi$ holds if either $\Box\varphi$ holds or $\varphi \mathcal{U} \psi$ holds.

3.4 Norms of Transmission

Norms of transmissions are expressed as temporal formulas. Each norm is either positive or negative. A positive norm might state that doctor Alice can send patient Charlie's test results to researcher Bob *if* Bob keeps the records in confidence. Negative norms are dual: they state communication can occur only if the temporal condition is satisfied. For example, doctor Alice can send patient Charlie's test results to researcher Bob *only if* Bob keeps the records in confidence. In the positive case, some other norm could authorize the communication and Bob would not be obliged to keep the results confidential, whereas in the negative case Bob must keep the results confidential regardless of how he obtained them from Alice.

We say a trace σ *satisfies the norms of context* c if Formula (1) of Fig. 1 holds. Formula (1) takes a disjunction over the *positive norms of transmission* for context c, denoted $\text{norms}^+(c)$, and a conjunction over the *negative norms of transmission* for context c, denoted $\text{norms}^-(c)$. Thus, in order to satisfy the norms, a communication must be allowed by at least one of the positive norms and it must respect all of the negative norms.

The syntactic forms of positive and negative norms are depicted in Fig. 1, where p_1, p_2, and q are variables of sort P, $\hat{r}_1$, $\hat{r}_2$, and $\hat{r}$ are terms of sort R, t is a variable of sort T, $\hat{t}$ is a term of sort T, θ is an *agent constraint*, and ψ is a *temporal condition*. An agent constraint θ is a formula free of temporal operators with free variables among p_1, p_2, and q. It expresses a relation among the sender, the recipient, and the subject, for example, that the sender and the subject are one and the same agent. A temporal condition ψ formalizes the notion of a principle of transmission and is a temporal formula with free variables among p_1, p_2, q, m, and t. It requires certain future actions to occur and certain past actions to have occurred (see Sect. 5 for concrete examples of norms).

One subtle consequence of the construction of Formula (1) is the treatment of attributes. Each individual norm applies to a downwardly closed set of attributes (downward in the information ordering on attributes induced by the computation rules). This captures the usual implication that the statement "allow disclosure of postal address" also allows the disclosure of postal codes. The formula universally quantifies over attributes because each communicated attribute must have a normative basis. The usual “upwards” inheritance of deny rules arises naturally here from the universal quantification over attributes and the downward closure of message contents. Suppose, for example, a norm denies the disclosure of postal codes. If one agent attempts to send a message containing a postal address, that message must also contain a postal code and when the attribute “postal code” is considered by the universal quantifier, the formula will forbid the disclosure.

4 Policies, Combination, and Compliance

A privacy policy regulates what flows of information are permitted between agents in various roles. A policy is a conjunction of contexts, requiring the norms of each context to be respected. For example, if Alice plays roles in both a bank and a hospital, she must act in accordance with the informational norms of both contexts.

Def. A *privacy policy* is a conjunction of formulas of the form (1) in Fig. 1.

We define below methods for evaluating privacy policies, both independently and in comparison with other policies. In addition, we define a notion of privacy compliance for an action. These problems can be solved using standard tools because they are formulated in LTL.

4.1 Consistency

A policy is consistent if it is possible for communicating agents to respect the policy. Inconsistent policies are not useful because they prescribe norms that agents cannot possibly satisfy. As defined, privacy policies can be satisfied trivially by agents who refrain from communicating any attributes. To focus on substantive consistency, we use a temporal formula, a *purpose*, to compel communication, requiring, for example, that eventually a bank customer receives his account balance.

Def. A privacy policy θ is *consistent* with a purpose α if there exists a trace σ such that $\sigma \models \theta \wedge \alpha$.

Because the satisfiability of LTL formulas is a well-studied problem, we can apply a set of known algorithmic results [39, 18, 29] to evaluate consistency of privacy policies. By assuming our carrier sets are finite, we are able to rewrite universal and existential quantifiers as finite conjunctions and disjunctions in Propositional LTL (PLTL).

Theorem 1. *Policy consistency can be decided in PSPACE.*

Let β be an LTL formula expressing the knowledge evolution constraints on traces. The proof idea is to propositionalize $\theta \wedge \alpha \wedge \beta$ and decide its satisfiability in PSPACE (with respect to formula length and the size of the carrier sets). Although the worst-case complexity of satisfiability is PSPACE, there are efficient algorithms for several syntactic classes of formulas [18]. Furthermore, there are tools that work well in practice, such as the widely used SPIN model-checker [24].

4.2 Entailment

Another metric for evaluating a privacy policy is to compare it against another policy. For example, a hospital’s privacy policy should not allow information flows prohibited by HIPAA.

Def. A privacy policy θ_1 *entails* a policy θ_2 if the LTL formula $\theta_1 \rightarrow \theta_2$ is valid over traces.

A hospital’s privacy policy should entail HIPAA (which in turn should entail the norms of the societal health care context). Entailment generalizes the notion of *policy refinement* defined for EPAL in [7, 9]. These previous definitions are lattice-theoretic and require direct reasoning about upwards and downwards inheritance. Our simpler model-theoretic definition is made possible by representing policies as logical formulas that properly quantify over attributes. Here, policy entailment reduces to standard logical implication.

Theorem 2. *Policy entailment can be decided in PSPACE.*

This theorem is proved by observing that the formula $\theta_1 \rightarrow \theta_2$ is valid over traces just in case $\neg(\theta_1 \rightarrow \theta_2) \wedge \beta$ is not satisfiable, where β is an LTL formula for knowledge constraints. Deciding policy entailment for our policies is more difficult than for other privacy languages because we directly model temporal constraints instead of abstracting them into uninterpreted “obligations” (see Sect. 6.3).

Policy entailment also leads to notions of *policy combination*, as in [10, 6]. Entailment as implication gives rise to combination as logical conjunction and disjunction. This replaces the previous complex lattice-based definitions of other privacy languages. Policy combination is simpler in this framework because we represent policies by carefully constructed logical formulas and not by functions, as in XACML and EPAL. Representing policies as functions loses essential information about whether a requirement was inherited from another attribute. Representing policies as logical formulas retains the inheritance information, simplifying combination.

4.3 Compliance

Finally, we address the issue of compliance: given the sequence of past communications, does the policy permit a contemplated communication and, if so, what future requirements are incurred? This question has both a weak and a strong formulation. The weak formulation requires the contemplated action to satisfy all the necessary present conditions imposed by the policy. These necessary conditions are tracked using a standard PLTL construction called the tableau [29]. The tableau of a PLTL formula is constructed by syntactically separating the present and future requirements. The future requirements characterize the sequences of actions that complete a finite trace to a satisfying infinite trace.

Def. Given a finite past history σ, an action a *weakly complies* with privacy policy θ if $\sigma \cdot a$ is a path in the tableau of θ that starts at an initial θ-atom. The *future requirements* of $\sigma \cdot a$ is the LTL formula ψ such that, for all traces σ',

$$\sigma' \models \psi \text{ if, and only if, } \sigma \cdot a \cdot \sigma' \models \theta.$$

Weak compliance ensures that each action taken by agents locally satisfies the privacy policy. However, a weakly compliant action could incur unsatisfiable future requirements. Weak compliance can be decided (and future requirements computed) using efficient techniques from LTL run-time verification [35].

Theorem 3. *Weak compliance and future requirements can be computed in polynomial time.*

In strong compliance, the information system ensures that agents can actually meet their future requirements while adhering to the policy. Note that previous privacy languages, such as EPAL, are able to determine only weak compliance because they lack a rich enough model of temporal conditions to determine the satisfiability of future requirements.

Def. Given a finite past history σ, an action a *strongly complies* with a privacy policy θ if there exists a trace σ' such that $\sigma \cdot a \cdot \sigma' \models \theta$.

Theorem 4. *Strong compliance can be decided in PSPACE.*

The complexity of checking strong compliance is in PSPACE because it involves checking for satisfiability. However, because the typical use of this algorithm will be at each point in a trace (for example in a hospital information system), it is natural to ask whether it is possible to reduce the complexity of checking whether each action is compliant by doing more work at the beginning of the execution. If weak compliance for a policy implies strong compliance, an information system need only require weak compliance (which can be computed efficiently) in order to achieve strong compliance.

Theorem 5. *Given a privacy policy θ, it can be decided whether weak compliance for θ implies strong compliance in exponential space.*

The main idea behind the proof is to construct the automaton for θ and check that there is a path from every reachable state to a strongly connected component.

5 Expressing Privacy Legislation

In this section, we exhibit the expressiveness of our formal model of contextual integrity by showing how to represent some commonly discussed privacy legislation. We intend our framework to express organizational privacy policies as well as legislation but focus on legislation in this section for concreteness. We can capture most of the privacy notions embedded in the laws we examine, and conversely the laws we examine exercise most of the features of our model. We regard this as evidence that CI has roughly the correct level of expressiveness to represent generally accepted notions of privacy.

We consider three pieces of legislation: the Health Insurance Portability and Accountability Act (HIPAA), the Children's Online Privacy Protection Act (COPPA), and the Gramm–Leach–Bliley Act (GLBA). The distinction between positive and negative norms surfaces in the different approaches taken by these laws. At a high level, HIPAA forbids disclosure of protected health information except in certain enumerated capacities, whereas COPPA and GLBA forbid certain enumerated information flows. Temporal conditions attached to negative norms are common in COPPA and GLBA. The mishandling of negative temporal conditions in other frameworks hampers their ability to capture these privacy laws correctly, whereas CI is able to capture both flavors of policy in a unified logical framework.

5.1 The HIPAA Privacy Rule

The HIPAA Privacy Rule regulates the transmission of "protected health information" (*phi*), by *covered entities*, such as hospitals, doctors, and insurance companies [31]. HIPAA largely forbids the disclosure of health information except to individuals or organizations acting in certain roles. HIPAA contains many privacy provisions, most of which can be expressed directly as positive transmission norms. We present a few representative examples in Fig. 2.

One norm (2) allows a covered entity to communicate *phi* about an individual to that individual. This norm allows Dr. Alice to show Bob an x-ray of his broken leg. It does not allow, however, Dr. Alice to show Bob's x-ray to Charlie. Moreover, it does not permit x-ray technician Debbie to give the x-ray to Dr. Alice. For that communication, HIPAA provides another norm (3). Dr. Alice is not only a covered entity, but more specifically a health care *provider*, someone directly involved in the care of a patient. Here, Debbie plays the role of covered entity and is permitted to give Bob's x-ray to Dr. Alice (Bob plays the role of patient).

Although the bulk of HIPAA consists of positive norms dealing with the attribute *phi*, HIPAA does contain a negative norm dealing with a component of *phi*: psychotherapy

$$\mathrm{inrole}(p_1, \textit{covered-entity}) \wedge \mathrm{inrole}(p_2, \textit{individual}) \wedge (q = p_2) \wedge (t \in \textit{phi}) \tag{2}$$

$$\mathrm{inrole}(p_1, \textit{covered-entity}) \wedge \mathrm{inrole}(p_2, \textit{provider}) \wedge \mathrm{inrole}(q, \textit{patient}) \wedge (t \in \textit{phi}) \tag{3}$$

$$\begin{aligned}&\mathrm{inrole}(p_1, \textit{covered-entity}) \wedge \mathrm{inrole}(p_2, \textit{individual}) \wedge (q = p_2) \wedge (t \in \textit{psychotherapy-notes}) \rightarrow \\ &\qquad \diamondminus \exists p : P.\, \mathrm{inrole}(p, \textit{psychiatrist}) \wedge \mathrm{send}(p, p_1, \textit{approve-disclose-psychotherapy-notes})\end{aligned} \tag{4}$$

$$\begin{aligned}&\mathrm{inrole}(p_1, \textit{covered-entity}) \wedge \mathrm{inrole}(p_2, \textit{individual}) \wedge \mathrm{inrole}(q, \textit{individual}) \wedge (t \in \textit{condition-and-location}) \wedge \\ &\qquad \diamondminus \exists m' : M.\, \mathrm{send}(p_2, p_1, m') \wedge \mathrm{contains}(m', q, \textit{name})\end{aligned} \tag{5}$$

$$\mathrm{inrole}(p_1, \textit{covered-entity}) \wedge \mathrm{inrole}(p_2, \textit{clergy}) \wedge \mathrm{inrole}(q, \textit{individual}) \wedge (t \in \textit{directory-information}) \tag{6}$$

Figure 2. Norms of Transmission from the HIPAA Privacy Rule

notes. The rule provides special protection for the disclosure of psychotherapy notes, even to the individual whom the notes are about. In particular, HIPAA contains a negative norm (4) that prevents a covered entity from disclosing psychotherapy notes to the subject of the notes without the prior approval of a psychiatrist.

The interplay between the positive and negative norms is subtle. One positive norm (2) permits the disclosure of psychotherapy notes, but a negative norm (4) prevents it (unless approval is obtained). These norms are not contradictory because the positive norm does not require the disclosure. Moreover, even after approval is received (satisfying the negative temporal condition), the covered entity would not be allowed to disclose the notes without the positive norm.

HIPAA contains specific norms for directories of facilities such as hospitals. Specifically, it provides that a covered entity may "disclose the individual's [general] condition and location within the facility to anyone asking for the individual by name" [31]. This can be expressed as the positive norm (5). The rule also contains a provision allowing members of the clergy to obtain directory information. This is expressed in norm (6), where *directory-information* is an attribute that contains (formally can be used to compute) the individual's name, general condition, religious affiliation, and location within the facility. The use of such information by the clergy is subject to further norms, but this is outside the scope of HIPAA.

De-identified Health Information. Most of the HIPAA privacy rule can be expressed using norms of transmission. Some provisions, however, fall outside our model. In particular, HIPAA provides that covered entities can disclose "de-identified health information" without regard to the other provisions of the rule. In our formalization of contextual integrity, every attribute is "identified" in virtue of being associated with an agent. Although we have not examined this in detail, we expect that an extended model with group attributes (attributes about a set of agents) could capture de-identified attributes. The relation between individual attributes and de-identified attributes has been studied extensively (e.g., [3, 20, 40]).

5.2 Children's Online Privacy Protection Act (COPPA)

COPPA protects the personal information children communicate to web sites [22]. It differs from HIPAA in two ways. First, COPPA does not contain an enumeration of positive norms. Instead, it contains two negative norms that restrict otherwise permissible flows of information. Second, temporal conditions play a central role in COPPA. The temporal conditions require web sites who collect protected information from children to respond in a certain way to messages from parents.

COPPA applies when a *child* sends individually identifiable information, *protected-info*, about him- or herself to a *web site* operator over the Internet. The two central negative norm of COPPA have a similar form, differing only in their temporal conditions. Whenever a child sends a web site his or her protected information, the web site operator is bound to follow both temporal conditions, one requiring "parental consent" and another providing a "right of access."

The negative norm (7) requires web site operators to obtain parental consent before collecting protected information from children. When a child sends protected information to a web site, a *parent* must have previously received a privacy notice from the web site operator, granted consent to the web site operator, and not since revoked that consent. Notice the strong form of "since" is required here to ensure that the parent actually granted consent. The negative norm (8) contains a temporal condition that requires web site operators to furnish parents with a privacy notice describing their information practices as well as the specific information they have collected from the child. This reactive condition is easily expressed using the $\Box\Diamond$ modality.

$$\begin{aligned}&\mathrm{inrole}(p_1, \textit{child}) \land \mathrm{inrole}(p_2, \textit{web-site}) \land (q = p_1) \land (t \in \textit{protected-info}) \to \\ &\qquad \exists p : P.\, \mathrm{inrole}(p, \textit{parent}) \land \neg\, \mathrm{send}(p, p_2, \textit{revoke-consent})\mathcal{S} \\ &\qquad\qquad (\mathrm{send}(p, p_2, \textit{grant-consent}) \land \Diamond \mathrm{send}(p_2, p, \textit{privacy-notice})) \quad (7)\end{aligned}$$

$$\begin{aligned}&\mathrm{inrole}(p_1, \textit{child}) \land \mathrm{inrole}(p_2, \textit{web-site}) \land (q = p_1) \land (t \in \textit{protected-info}) \to \\ &\qquad \Box \forall p : P.\, \mathrm{inrole}(p, \textit{parent}) \land \mathrm{send}(p, p_2, \textit{request-information}) \to \\ &\qquad\qquad \Diamond(\mathrm{send}(p_2, p, \textit{privacy-notice}) \land \mathrm{send}(p_2, p, m)) \quad (8)\end{aligned}$$

Figure 3. Norms of Transmission from COPPA

The first temporal condition is concerned with the past, that a parent has given consent, whereas the second condition is concerned with the future, that the web site operator reacts correctly to parental requests. COPPA requires web site operators to verify that they are indeed communicating with one of the child's parents before disclosing the child's protected information. Such verification is represented in our model by assigning the role *parent* to the appropriate agents. COPPA also requires the operator to delete protected information in its possession upon receiving *revoke-consent*. Our model does not capture "forgetting" actions, but such actions can be included in the model, at the cost of complexity.

5.3 Gramm–Leach–Bliley Act (GLBA)

The Financial Modernization Act of 1999, commonly referred to as the Gramm–Leach–Bliley Act or GLBA, contains privacy provisions limiting how financial institutions can handle the non-public personal information, *npi*, of their customers and consumers [23]. Broadly, GLBA requires financial institutions to inform their customers of their privacy practices and to allow customers to "opt-out" of certain kinds of information disclosures.

Financial *institutions* are required to send their *customers* privacy notices every year as long the customer relationship lasts. Without numerical notions of time, CI can not express that the notices must be delivered annually. Instead, the negative norm (9) requires institutions to periodically send privacy notices.

In addition to a customer role, GLBA distinguishes a *consumer* role. GLBA's requirements on interacting with consumers are less strict than its requirements on interacting with customers. Institutions are required to notify consumers of their privacy practices only if they share the consumer's *npi* with *non-affiliated* companies, and they may do so before or after the disclosing *npi*. The negative norm (10) makes essential use of the three different roles (sender, recipient, and subject), as well as both past and future modalities in its temporal condition.

Both consumers and customers can "opt-out" of the sharing of *npi* with non-affiliated companies. The norm (11) expresses the provision for consumers, and GLBA also contains an analogous non-affiliate opt-out norm for customers. Consumers and customers also have the option of opting out of some kinds information sharing between institutions and their *affiliates*, such the sharing of credit reports and application information. The norm (12) expresses the provision, and GLBA contains a similar norm for application information. GLBA contains some exceptions to these norm, but we omit those here for clarity.

Much of the consternation about GLBA revolves around the complex definition of which companies are affiliates and what precisely constitutes non-public personal information [21]. Our formalization of these norms sidesteps these issues by taking the role *affiliate* and the attribute *npi* to be defined exogenously: the judgments as to which companies are *affiliates* and which communications contain *npi* are made in the preparation of a trace history. The machinery of the model then classifies this trace history as respecting or as not respecting the norms of transmission.

The use of negative norms in the expression of GLBA is essential: replacing the negative norms with their positive duals fails to express GLBA. Consider Alice, who is both a customer and a consumer of financial institution FirstCyber. In the negative formulation of GLBA, if she sends *npi* to FirstCyber, FirstCyber must periodically send her privacy notices. In the attempted positive formulation, however, if she sends *npi* to FirstCyber, FirstCyber need not periodically send her privacy notices. The disjunctive character of positive norms enables FirstCyber to choose, for each communication, whether to regard Alice as a customer or as a consumer. In the negative formulation, the conjunctive character of the negative norms requires FirstCyber to treat Alice as both a customer and a consumer.

6 Comparison with Other Models

In this section, we compare CI with traditional Role-Based Access Control (RBAC), the eXtensible Access Con-

$$\begin{aligned}&\mathrm{inrole}(p_1, \textit{customer}) \wedge \mathrm{inrole}(p_2, \textit{institution}) \wedge (q = p_1) \wedge (t \in \textit{npi}) \rightarrow \\ &\qquad \Diamond\mathrm{send}(p_2, p_1, \textit{privacy-notice})\mathcal{W}\neg\,\mathrm{inrole}(p_1, \textit{customer}) \qquad (9)\end{aligned}$$

$$\begin{aligned}&\mathrm{inrole}(p_1, \textit{institution}) \wedge \mathrm{inrole}(p_2, \textit{non-affiliate}) \wedge \mathrm{inrole}(q, \textit{consumer}) \wedge (t \in \textit{npi}) \rightarrow \\ &\qquad \Diamond\mathrm{send}(p_1, q, \textit{privacy-notice}) \vee \diamondminus\mathrm{send}(p_1, q, \textit{privacy-notice}) \qquad (10)\end{aligned}$$

$$\begin{aligned}&\mathrm{inrole}(p_1, \textit{institution}) \wedge \mathrm{inrole}(p_2, \textit{non-affiliate}) \wedge \mathrm{inrole}(q, \textit{consumer}) \wedge (t \in \textit{npi}) \rightarrow \\ &\qquad \neg\diamondminus\mathrm{send}(q, p_1, \textit{opt-out-of-non-affiliate}) \qquad (11)\end{aligned}$$

$$\begin{aligned}&\mathrm{inrole}(p_1, \textit{institution}) \wedge \mathrm{inrole}(p_2, \textit{affiliate}) \wedge \mathrm{inrole}(q, \textit{consumer}) \wedge (t \in \textit{credit-report}) \rightarrow \\ &\qquad \neg\diamondminus\mathrm{send}(q, p_1, \textit{opt-out-of-affiliate}) \qquad (12)\end{aligned}$$

Figure 4. Norms of Transmission from GLBA

trol Markup Language (XACML), the Enterprise Privacy Authorization Language (EPAL), and the Platform for Privacy Preferences (P3P). CI generalizes these existing models in two key ways. First, CI includes an extensive language for defining temporal conditions, improving the rudimentary future "obligations" of XACML and EPAL. Second, CI correctly handles temporal conditions associated with negative norms (denying rules). Temporal conditions can be attached to denying rules in XACML and EPAL, but the resulting semantics are murky. Our findings are summarized in Figure 5.

6.1 Role-Based Access Control

Role-Based Access Control (e.g., [12]) is an access control model in which access rights are specified in terms of roles. CI generalizes RBAC by specifying more parameters by roles, containing a notion of attribute and data subject, and including temporal conditions. RBAC can express policies about arbitrary actions, whereas CI, as currently formulated, is concerned solely with communication actions. CI replaces the "object" of RBAC with a recipient principal, enabling the "actee" (object or recipient) to be specified by a role. RBAC rules are positive and negative norms of the following forms, respectively:

$$\begin{aligned}\text{Allow:}\quad & \mathrm{inrole}(p_1, \hat{r}_1) \wedge (p_2 = \hat{p}_2) \\ \text{Deny:}\quad & \mathrm{inrole}(p_1, \hat{r}_1) \wedge (p_2 = \hat{p}_2) \rightarrow \bot\end{aligned}$$

Notice RBAC lacks the subject q and attribute t. Temporal conditions are also absent. "Deny" rules are expressible in CI by negative norms with $\bot$, the unsatisfiable formula.

The key reason RBAC is insufficient for privacy is it lacks the notion of an attribute. Suppose a doctor reads a patient's medical file and then sends an email to his broker. From an RBAC perspective, nothing untoward has occurred. Both actions, reading the file and sending the email, are (presumably) permitted by the policy. However, a privacy breach has occurred if the doctor includes sensitive medical information about another patient in his email. To distinguish the appropriate from the inappropriate, it is essential to recognize the attributes communicated by each action. In other words, RBAC is insufficient for privacy because it lacks the "$\mathrm{contains}$" relation.

Several access control languages, such as Binder [19] and RT [28], extend RBAC using Datalog. Typically, these languages use only positive rules and contain neither temporal conditions nor a notion of the subject of a piece of information. Cassandra [11], a sophisticated access control language with denying rules, has been applied to electronic health records in the United Kingdom. In that study, consent was captured through role activation: a patient consents to treatment by activating a "consent-to-treatment" role. Future temporal constraints, as well as notions of computing attributes, are absent.

6.2 Extensible Access Control Markup Language

The Extensible Access Control Markup Language [5] is a flexible language for expressing access control policies. XACML's extension mechanism enables XACML to capture a wide variety of access control constructs. To make meaningful statements about the expressiveness of XACML, we restrict our attention to policies expressible by simple extensions to the base XACML language. In particular, we abstract XACML's targets as elements of a Boolean algebra over a set of requests and consider only the built-in combination algorithms.

XACML lacks first-class temporal conditions. When an XACML policy reaches a policy judgment, it can include in its response an "obligation," a symbol to be interpreted at the point of policy enforcement. These uninterpreted symbols can be used to represent future requirements. Obliga-

Model	Sender	Recipient	Subject	Attributes	Past	Future	Combination
RBAC	Role	Identity	×	×	×	×	•
XACML	Flexible	Flexible	Flexible	○	×	○	•
EPAL	Fixed	Role	Fixed	•	×	○	×
P3P	Fixed	Role	Fixed	•	○	×	○
CI	Role	Role	Role	•	•	•	•

Figure 5. Comparison of various privacy languages. The symbol × indicates the feature is absent from the language, ○ indicates partial or limited functionality, and • indicates the feature is fully functional. Note, [6] gives an extension of EPAL that is closed under combination.

tions, however, prevent the semantics of an XACML policy from being fully specified by the policy itself (as the policy relies on the surrounding environment to give the obligations meaning). Past conditions can also be expressed in XACML by encoding state information into the "request context," additional information passed to the policy evaluation engine. However, using this feature to capture state more complex than "opt-in" and "opt-out" is awkward.

XACML is unable to correctly capture attributes [4], especially in connection with denying rules (negative norms). The difficulty arises because XACML conceives of a policy as a *function* from requests to responses. XACML policies are structured as combinations of simple subpolicies, where combination is computed point-wise on the functions represented by the subpolicies. This fails for attributes because the effect of combination can be non-local (due to "upward" inheritance). The combined response for two policies on a request is not necessarily determined by the responses of the subpolicies on *that* request. CI avoids this by representing and combining policies logically.

6.3 The Enterprise Privacy Authorization Language

The Enterprise Privacy Authorization Language is expressly designed for expressing enterprise privacy policies [8, 38]. EPAL policies are concerned with a single sender (the enterprise itself) and a single subject role [27]. EPAL has the same limitations as XACML on its temporal conditions.

EPAL requests are elements of a Cartesian product of trees representing roles, attributes, purposes, and actions. The "role" coordinate represents the role of the recipient. The "purpose" coordinate is not captured directly in CI. However, these purposes can be simulated in CI (see below). Finally, EPAL policies are concerned with general actions, not just with communication actions, as in RBAC. With the exception of purposes and non-communication actions, CI captures EPAL policies using positive and negative norms of the following forms, respectively:

$$(p_1 = \hat{p}_1) \wedge \text{inrole}(p_2, \hat{r}_2) \wedge (t \in \hat{t}) \wedge \hat{o}$$
$$(p_1 = \hat{p}_1) \wedge \text{inrole}(p_2, \hat{r}_2) \wedge (t \in \hat{t}) \rightarrow \bot$$

The sender agent $\hat{p}_1$ is fixed for every norm in a single policy. The symbol $\hat{o}$ is a propositional letter that represents an uninterpreted future "obligation," similar to those found in XACML. EPAL structures these obligations with a subsumption relation.

CI improves on EPAL obligations in two respects. First, obligations are expressed in temporal logic (as in [26]), the same logic as the policy itself. Thus, tools can interpret temporal conditions, and determine, for example, whether or not it is possible for an agent to discharge his or her future obligations while adhering to the policy. Second, our temporal conditions can speak about the past as well as the future, enabling policies which permit information flows in virtue of past actions. In CI, the subsumption relation on temporal conditions arises naturally as logical implication of temporal formulas. Future obligations in the form of a list of future actions that must be performed are present in the policy specification language Ponder [17]. These obligations are richer than EPAL's uninterpreted obligations, but are restricted to $\Diamond$ conditions, failing to capture the reactive $\Box\Diamond$ condition in COPPA norm (8), for example.

EPAL policy authors can attach obligations to denying rules. However, the semantics of such obligations are dubious: the policy engine responds that a contemplated action is both denied and incurs an obligation, but is the obligation incurred if the requesting agent does not perform the contemplated action? CI resolves this difficulty by weakening the notion of a denying rule to that of a negative norm, a formula of the form $\varphi \rightarrow \psi$. Negative norms do not forbid actions described by φ, but instead forbid actions described by φ that violate the temporal condition ψ. Complete prohibitions can be expressed by instantiating ψ with $\bot$.

EPAL purposes in CI. In EPAL, each action is conducted for some purpose. An EPAL policy can permit an action for a particular purpose and also deny the same action for a different purpose. For example, a health web site might be permitted to analyze visitor health information for medical purposes, but might not be permitted to analyze the same health information for marketing purposes. CI can capture this notion by decomposing large agents into several smaller agents, one for each purpose. For example, the monolithic health web site could be decomposed into a medical agent and a marketing agent. EPAL purposes could then be expressed in CI by restricting communication among the constituent agents.

6.4 Platform for Privacy Preferences

The Platform for Privacy Preferences (P3P) is a privacy language intended for use by web site operators in informing their visitors of their data practices [16, 34]. P3P contains only positive norms and very restricted temporal conditions. Additionally, a single P3P policy is restricted to a single sender (the web site) and a single subject role (a web site visitor). These restrictions impair the use of P3P as a general-purpose privacy language. For example, P3P is unable to directly express that a web site conforms to COPPA. To make such a statement, a web site operator must employ a P3P extension [15] and make the policy statement `COPPA status="compliant"`. Temporal conditions in P3P are limited to `opt-in`, `opt-out`, and `true`. P3P statements correspond to positive norms of the following form:

$$(p_1 = \hat{p}_1) \land \mathrm{inrole}(p_2, \hat{r}_2) \land \mathrm{inrole}(q, \textit{visitor}) \land (t \in \hat{t}) \land \psi$$

where ψ represents “opt-in,” “opt-out,” or no temporal condition. The lack of negative norms simplifies P3P at the cost of expressiveness. The fixed form of the opt-in and opt-out conditions is restrictive, preventing even minor variations such as the parental “grant-consent” and “revoke-consent” idiom found in COPPA.

P3P provides for privacy preference languages that a web surfer can use to filter out web sites with unwanted data practices. These preferences languages highlight another difference between P3P and CI: all P3P policies inhabit a single global context. A web surfer cannot specify different preferences for medical web sites than for financial web sites. This forces web surfers to resort to a “lowest common denominator” preference. Both the preference languages APPEL [16] and XPref [2] can express negative preferences, but such preferences are not respected in the full P3P system [9].

7 Conclusions and Future Work

We present a logical framework for expressing and reasoning about norms of transmission of personal information. This framework formalizes some central ideas of contextual integrity, a conceptual framework for understanding privacy expectations that has been developed in the literature on law and public policy. Privacy norms are expressed as LTL formulas and interpreted over traces in which the basic actions include communication of the form “Alice gives Bob a certain type of information about Charlie.” A basic premise of contextual integrity is that appropriateness depends on the context, the role, and the subject of personal information, and cannot be captured accurately using a DRM-style “ownership of information” model or a simple partitioning of information into “public information” and “private information.” We illustrate the use of the framework by showing how to express several privacy provisions of HIPAA, COPPA, and GLBA as temporal logic formulas about the transmission of personal information.

We show that questions of policy consistency, compliance, combination, and refinement reduce to well-studied problems in LTL. Policy combination, which has proven problematic in EPAL, is formulated easily using logical conjunction and disjunction, and policy refinement similarly reduces to logical implication. In deciding compliance, we are given a sequence of past communications and wish to determine whether a possible next communication will violate the privacy policy. This has both weak and strong formulation: weak compliance requires only that the next action satisfies all necessary present conditions, whereas strong compliance requires, in addition, that there is an achievable sequence of future actions that meets all requirements about the future. Weak compliance can be checked in polynomial time using results from runtime verification, whereas strong compliance checks require PSPACE complexity.

We compare our framework to previous access control and privacy policy languages including RBAC, XACML, EPAL, and P3P. Our results are summarized in Fig. 5. In particular, the two central concepts of our framework—temporal conditions and separation between positive and negative norms—seem to capture essential features used in writing privacy policies. Specifically, temporal conditions improve on the uninterpreted future obligations of XACML and EPAL, and the use of negative norms obviates the problems with obligations attached to denying rules in previous frameworks.

Future Work. In future work, we hope to apply the model by using a model checker to analyze whether privacy norms contained in the HIPAA privacy rule are consistent with certain purposes and entail specific desired properties. We also hope to apply the model in a system for handling electronic health records to ensure that the system complies with HIPAA.

Currently, our framework assumes that norms are based only on the type of information (rather than actual data values) and that information is about a single individual (rather than about a group of individuals). We plan to extend the formalization by relaxing these restrictions, enabling norms to depend on specific data values and information to describe groups of individuals. In this extended framework, we hope to develop precise connections with research on data privacy and aggregation.

We also plan to extend the framework to include parameterized roles. These parameterized roles would enable CI to capture certain norm more precisely. For example, norm (7) could be expressed more precisely with a parameterized parent role, ensuring that the consenting parent is actually the child's parent. Parameterized roles are present in other policy languages, such as RT [28], and are appropriate for privacy languages.

Finally, our current language faces a limitation common to many policy languages. Consider SB 1386, a California law requiring businesses that inappropriately disclose personal information to notify the subjects of the information. This provision cannot be expressed properly in the language because it takes effect only when an agent violates norms. In our model, agents never violate norms and thus would never be required to notify individuals. However, such notifications are common in California. To express such "defense in depth" provisions, we plan to extend our model to account for agents who occasionally (perhaps unintentionally) violate the norms. We expect this to require modifications to the current logic.

Acknowledgments. We thank A. Bradley, C. Sánchez, and M. Slanina for helpful discussions about LTL. This work was partially supported by the NSF Cybertrust Program through the PORTIA project and the NSF Science and Technology Center TRUST.

References

[1] M. S. Ackerman, L. F. Cranor, and J. Reagle. Privacy in e-commerce: examining user scenarios and privacy preferences. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 1–8. ACM Press, 1999.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. An XPath-based preference language for P3P. In *Proceedings of the Twelfth International Conference on World Wide Web*, pages 629–639. ACM Press, 2003.

[3] R. Agrawal, R. Srikant, and D. Thomas. Privacy preserving OLAP. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 251–262, New York, NY, USA, 2005. ACM Press.

[4] A. Anderson. Key differences between XACML and EPAL. Ottawa new challenges for access control, 2005.

[5] A. Anderson, A. Nadalin, B. Parducci, D. Engovatov, E. Coyne, F. Siebenlist, H. Lockhart, M. McIntosh, M. Kudo, P. Humenn, R. Jacobson, S. Proctor, S. Godik, S. Anderson, and T. Moses. Extensible access control markup language (XACML) version 2.0, 2004.

[6] M. Backes, M. Dürmuth, and R. Steinwandt. An algebra for composing enterprise privacy policies. In *European Symposium on Research in Computer Security (ESORICS)*, volume 3193 of *LNCS*. Springer–Verlag, 2004.

[7] M. Backes, G. Karjoth, W. Bagga, and M. Schunter. Efficient comparison of enterprise privacy policies. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 375–382. ACM Press, 2004.

[8] M. Backes, B. Pfitzmann, and M. Schunter. A toolkit for managing enterprise privacy policies. In *European Symposium on Research in Computer Security (ESORICS)*, volume 2808 of *LNCS*, pages 101–119. Springer–Verlag, 2003.

[9] A. Barth and J. C. Mitchell. Enterprise privacy promises and enforcement. In *WITS '05: Proceedings of the 2005 Workshop on Issues in the Theory of Security*, pages 58–66, New York, NY, USA, 2005. ACM Press.

[10] A. Barth, J. C. Mitchell, and J. Rosenstein. Conflict and combination in privacy policy languages. In *Proceedings of the 2004 Workshop on Privacy in the Electronic Society*. ACM Press, 2004.

[11] M. Y. Becker and P. Sewell. Cassandra: Flexible trust management, applied to electronic health records. In *CSFW '04: Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW'04)*, page 139, Washington, DC, USA, 2004. IEEE Computer Society.

[12] M. Bishop. *Computer Security: Art and Science*. Addison Wesley Professional, 2003.

[13] S. Byers, L. F. Cranor, and D. Kormann. Automated analysis of P3P-enabled web sites. In *Proceedings of the 5th International Conference on Electronic Commerce*, pages 326–338. ACM Press, 2003.

[14] J. Crampton. On permissions, inheritance and role hierarchies. In *Proceedings of the 10th ACM Conference on Computer and Communication Security*, pages 85–92. ACM Press, 2003.

[15] L. F. Cranor. *Web Privacy with P3P*. O'Reilly and Associates, Inc., 2002.

[16] L. F. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The platform for privacy preferences 1.0 (P3P1.0) specification, 2002. http://www.w3.org/TR/P3P/.

[17] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The ponder policy specification language. In *POLICY '01: Proceedings of the International Workshop on Policies for Distributed Systems and Networks*, pages 18–38, London, UK, 2001. Springer-Verlag.

[18] S. Demri and P. Schnoebelen. The complexity of propositional linear temporal logics in simple cases. In *Procceding of the 15th Annual Symposium on Theoretical Aspects of Computer Science (STACS'98)*, volume 1373 of *LNCS*. Springer–Verlag, 1998.

[19] J. DeTreville. Binder, a logic-based security language. In *SP '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy*, page 105, Washington, DC, USA, 2002. IEEE Computer Society.

[20] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *CRYPTO 2004: 24th Annual International Cryptology Conference*, volume 3152 of *LNCS*, pages 528–544. Springer–Verlag, 2004.

[21] Electronic Privacy Information Center. The Gramm–Leach–Bliley Act, 2005. http://www.epic.org/privacy/glba/.

[22] Federal Trade Commission. How to comply with the children's online privacy protection rule, 1999. http://www.ftc.gov/bcp/conline/pubs/buspubs/coppa.htm.

[23] Federal Trade Commission. In brief: the financial privacy requirements of the Gramm–Leach–Bliley Act, 2002. http://www.ftc.gov/bcp/conline/pubs/buspubs/glbshort.htm.

[24] G. J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison Wesley Professional, 2004.

[25] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Trans. Database Syst.*, 26(2):214–260, 2001.

[26] W. Jamroga, W. van der Hoek, and M. Wooldridge. On obligations and abilities. In *Deontic Logic: 7th International Workshop on Deontic Logic in Computer Science*, volume 3065 of *LNCS*, pages 165–181. Springer–Verlag, 2004.

[27] G. Karjoth and M. Schunter. A privacy policy model for enterprises. In *15th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, 2002.

[28] N. Li and J. C. Mitchell. RT: A role-based trust-management framework. In *The Third DARPA Information Survivability Conference and Exposition*, pages 201–212, Washington, DC, USA, 2003. IEEE Computer Society.

[29] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer–Verlag, 1995.

[30] H. Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–158, 2004.

[31] Office for Civil Rights. Summary of the HIPAA privacy rule. United States Department of Health & Human Services, 2003.

[32] J. Prins. The propertization of personal data and identities. *Electronic Journal of Comparative Law*, 8.3, October 2004.

[33] J. Rachels. Why privacy is important. In F. D. Schoeman, editor, *Philosophical Dimensions of Privacy: An Anthology*, pages 290–294. 1984.

[34] J. Reagle and L. F. Cranor. The platform for privacy preferences. *Communications of the ACM*, 42(2):48–55, 1999.

[35] G. Rosu and K. Havelund. Synthesizing dynamic programming algorithms for linear temporal logic formulae. Technical Report TR 01-15, RIACS, May 2001.

[36] F. Schoeman. Privacy and intimate information. In F. D. Schoeman, editor, *Philosophical Dimensions of Privacy: An Anthology*, pages 403–408. 1984.

[37] F. Schoeman. Gossip and privacy. In R. F. Goodman and A. Ben-Zeev, editors, *Good Gossip*, pages 403–408. 1994.

[38] M. Schunter, P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter. Enterprise privacy authorization language (EPAL 1.1), 2003. http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/.

[39] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, July 1985.

[40] L. Sweeney. k-Anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

A Temporal logic

Syntax. Formulas free of temporal modalities refer to single states in the trace history of the agent world. Our sorts are P, T, M, R, and C (denoting agents, attributes, messages, roles, and contexts), with carriers $\mathcal{P}$, $\mathcal{T}$, $\mathcal{M}$, $\mathcal{R}$, and $\mathcal{C}$, respectively. Our relations are as follows:

$$\begin{array}{rlrl} \text{send} : & P \times P \times M & \text{contains} : & M \times P \times T \\ \text{inrole} : & P \times R & \text{incontext} : & P \times C \\ \in : & T \times T. & & \end{array}$$

Intuitively, $\text{send}(p_1, p_2, m)$ holds in a state of a trace if agent p_1 just sent message m to agent p_2; $\text{contains}(m, q, t)$ holds if message m contains the value of attribute t of agent q; $\text{inrole}(p, r)$ holds in a state if agent p is active in role r; $\text{incontext}(p, c)$ holds in a state if agent p is active in a role of context c; Finally, $t \in t'$ holds if attribute t can be computed from (is a component of) attribute t'.

To generate the syntax of the logic, we use a sort assignment Γ. that records the sorts of variables bound by quantifiers. The recursive definition ensures that variables are used only as permitted by their sort. Formally, the set of terms $\text{Terms}^\tau(\Gamma)$ of sort τ under sort assignment Γ is generated as follows:

$$\begin{array}{cc} p \in \text{Terms}^P(\Gamma) & t \in \text{Terms}^T(\Gamma) \\ m \in \text{Terms}^M(\Gamma) & r \in \text{Terms}^R(\Gamma) \\ c \in \text{Terms}^C(\Gamma) & x \in \text{Terms}^\tau(\Gamma) \end{array}$$

for all $p \in \mathcal{P}$, $t \in \mathcal{T}$, $m \in \mathcal{M}$, $r \in \mathcal{R}$, $c \in \mathcal{C}$, and $x : \tau \in \Gamma$. Notice we have constant symbols for each element of each carrier. Formulas for sort assignment Γ are generated using the following grammar:

$$\begin{aligned} \varphi^\Gamma ::= {} & \text{send}(p_1, p_2, m) \mid \text{contains}(m, q, t) \mid \\ & \text{inrole}(p, r) \mid \text{incontext}(p, c) \mid t \in t' \mid \\ & \varphi^\Gamma \wedge \varphi^\Gamma \mid \neg\varphi^\Gamma \mid \varphi^\Gamma \mathcal{U} \varphi^\Gamma \mid \varphi^\Gamma \mathcal{S} \varphi^\Gamma \mid \\ & \bigcirc\varphi^\Gamma \mid \exists x : \tau . \varphi^{\Gamma'} \end{aligned}$$

where $p_1, p_2, q \in \text{Terms}^P(\Gamma)$, $m \in \text{Terms}^M(\Gamma)$, $t, t' \in \text{Terms}^T(\Gamma)$, $r \in \text{Terms}^R(\Gamma)$, $c \in \text{Terms}^C(\Gamma)$, x is a variable, τ is a sort, and Γ' is the sort assignment that agrees with Γ on all variables except x, to which Γ' assigns sort τ. We also include equality, defined in the usual manner.

Intuitively, $\varphi \mathcal{U} \psi$ holds just in case φ holds until ψ holds, ψ will eventually hold. The modality "since," written $\mathcal{S}$ is the past version of $\mathcal{U}$. The formula $\varphi \mathcal{S} \psi$ holds just in case φ has held since ψ held, and ψ has held in the past. $\bigcirc\varphi$ holds just in case φ holds in the next state. Finally, $\exists$ is rigid existential quantification.

Semantics. An *environment* is a function η from variables to $\mathcal{P} \cup \mathcal{T} \cup \mathcal{M} \cup \mathcal{R} \cup \mathcal{C}$. We write $\eta \models \Gamma$ if, for all $x : \tau \in \Gamma$, $\eta(x) \in A^\tau$, where A^τ is the carrier for sort τ. If $x \in \text{Terms}^\tau(\Gamma)$ and $\eta \models \Gamma$,

$$[\![x]\!]\eta = \begin{cases} \eta(x) & \text{if } x : \tau \in \Gamma, \\ x & \text{otherwise.} \end{cases}$$

For all infinite traces $\sigma = (\kappa_0, \rho_0, a_0), (\kappa_1, \rho_1, a_1), \ldots$ and all environments $\eta \models \Gamma$ such that $p_1, p_2, q \in \text{Terms}^P(\Gamma)$, $t, t' \in \text{Terms}^T(\Gamma)$, $m \in \text{Terms}^M(\Gamma)$, $r \in \text{Terms}^R(\Gamma)$, and $c \in \text{Terms}^C(\Gamma)$,

$$\begin{aligned}
&\sigma, i, \eta \models \text{send}(p_1, p_2, m) \\
&\qquad \iff a_i = ([\![p_1]\!]\eta, [\![p_2]\!]\eta, [\![m]\!]\eta) \\
&\sigma, i, \eta \models \text{contains}(m, q, t) \\
&\qquad \iff ([\![q]\!]\eta, [\![t]\!]\eta) \in \text{content}([\![m]\!]\eta) \\
&\sigma, i, \eta \models \text{inrole}(p, r) \\
&\qquad \iff ([\![p]\!]\eta, [\![r]\!]\eta) \in \rho_i \\
&\sigma, i, \eta \models \text{incontext}(p, c) \\
&\qquad \iff \text{exists } r \in [\![c]\!]\eta \text{ such that } ([\![p]\!]\eta, r) \in \rho_i \\
&\sigma, i, \eta \models t \in t' \\
&\qquad \iff (\{[\![t']\!]\eta\}, [\![t]\!]\eta) \text{ is a computation rule}
\end{aligned}$$

We extend $\models$ to formulas in the usual manner:

$$\begin{aligned}
&\sigma, i, \eta \models \varphi_1 \wedge \varphi_2 \\
&\qquad \iff \sigma, i, \eta \models \varphi_1 \text{ and } \sigma, i, \eta \models \varphi_2 \\
&\sigma, i, \eta \models \neg\varphi \\
&\qquad \iff \sigma, i, \eta \not\models \varphi \\
&\sigma, i, \eta \models \varphi_1 \mathcal{U} \varphi_2 \\
&\qquad \iff \text{exists } k \geq i \text{ such that } \sigma, k, \eta \models \varphi_2 \text{ and,} \\
&\qquad\qquad \text{for all } j,\, i \leq j < k \text{ implies } \sigma, j, \eta \models \varphi_1 \\
&\sigma, i, \eta \models \varphi_1 \mathcal{S} \varphi_2 \\
&\qquad \iff \text{exists } k \leq i \text{ such that } \sigma, k, \eta \models \varphi_2 \text{ and,} \\
&\qquad\qquad \text{for all } j,\, i \geq j > k \text{ implies } \sigma, j, \eta \models \varphi_1 \\
&\sigma, i, \eta \models \bigcirc\varphi_1 \\
&\qquad \iff \sigma, i+1, \eta \models \varphi_1 \\
&\sigma, i, \eta \models \exists x : \tau.\varphi \\
&\qquad \iff \text{exists } a \in A^\tau \text{ such that } \sigma, i, \eta[x \to a] \models \varphi
\end{aligned}$$

A^τ is the carrier of sort τ, and $\eta[x \to a]$ is the environment that agrees with η on all variable except x, where $\eta[x \to a]$ takes on value a.

Notation. To simplify notation, we use the following standard symbols:

$$\begin{array}{ll}
\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2) & \varphi_1 \to \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2 \\
\Diamond\varphi \equiv \top\mathcal{U}\varphi & \Box\varphi \equiv \neg\Diamond\neg\varphi \\
\diamondminus\varphi \equiv \top\mathcal{S}\varphi & \boxminus\varphi \equiv \neg\diamondminus\neg\varphi \\
\varphi_1\mathcal{W}\varphi_2 \equiv \varphi_1\mathcal{U}\varphi_2 \vee \Box\varphi_1 & \varphi_1\mathcal{B}\varphi_2 \equiv \varphi_1\mathcal{S}\varphi_2 \vee \boxminus\varphi_1 \\
\forall x : \tau.\varphi \equiv \neg\exists x : \tau.\neg\varphi &
\end{array}$$

The formula $\Diamond\varphi$ is read "eventually φ," and indicates that φ will eventually hold. Its dual modality, $\Box$, is read "henceforth." The modalities $\diamondminus$ and $\boxminus$ are the past forms of $\Diamond$ and $\Box$, respectively. We will often write $\sigma \models \varphi$ in place of $\sigma, 0, \eta \models \varphi$ when φ has no free variables (and thus does not depend on η).