Contextual Integrity for Argumentation-based Privacy Reasoning

Gideon Ogunniye  
Department of Science, Technology, Engineering and Public Policy (UCL STEaPP), University College London, London, United Kingdom  
g.ogunniye@ucl.ac.uk

Nadin Kökciyan  
School of Informatics, University of Edinburgh, Edinburgh, United Kingdom  
nadin.kokciyan@ed.ac.uk

ABSTRACT
Privacy management in online systems is a complex task. Recently, contextual integrity theory has been introduced to model privacy, which considers the social contexts of users before making privacy decisions. However, having a practical application based on this theory is not straightforward. In this paper, we propose an agent-based framework for privacy policy reasoning that combines the power of ontologies together with argumentation techniques to resolve privacy conflicts. First, we propose an ontology that represents the contextual integrity theory. We then introduce an argumentation-based dialogue framework that could: (i) reason about contextual norms to resolve privacy conflicts among agents, and (ii) provide justifications to the agents during multi-party dialogues. We apply our approach to privacy scenarios in various contexts where each scenario has different challenges to address. We conclude with theoretical results to show the effectiveness of the framework.

KEYWORDS
Privacy; Contextual integrity; Privacy ontology; Argumentation

ACM Reference Format:

1 INTRODUCTION
Online systems are used by billions of users who interact with each other, share data about themselves and also other people. Recent literature shows that online social network (OSN) users share personal information that could lead to privacy breaches [13, 17, 35]. On the other hand, OSNs have become one of the most disruptive communication platforms with high socioeconomic value [22]. For example, sharing content with an unintended audience could result in discriminatory practices [13]. The situation becomes even worse in the IoT domain, where users face the challenge of making privacy decisions in unseen situations [20]. Clearly, online systems have become platforms where users are vulnerable; and there is a need for tools that could help the users to manage their privacy.

Privacy is handled differently in various systems. In general, each piece of content to be shared is associated with a privacy policy that dictates who can access the content in question. One can provide a specific audience for a post (e.g., Facebook); or the content can be shared publicly or privately (e.g., Twitter). Managing privacy becomes more difficult when multiple people are involved in a content (e.g., a picture representing a group of people) [7, 8, 15]. In current OSNs, users do not have any control about setting multi-party policies; and they can only define a privacy policy of a content they are willing to share themselves (i.e., the content owners). Sharing a multi-party content without consulting the parties involved will most likely result in a multi-party privacy conflict (MPPC) [15, 25]. Furthermore, privacy is very subjective; agreeing on a common privacy policy is not trivial and requires automatic mechanisms to facilitate such negotiations to resolve MPPCs. Kokciyan, Yaglicli and Yolum propose a framework where agents, each representing an OSN user, agree on a sharing decision by using assumption-based argumentation [15]. Such and Rovatsos propose an approach in which each user could assign different degrees of privacy to set the audience of a content [36]. However, privacy is not only about access control. Unlike previous work, we develop a new agent framework in which privacy could be defined individually based on a well-formed theory, namely Contextual Integrity. This could help us develop personal privacy assistants (e.g., agents) to assist users in their decision-making based on specific contexts.

In Contextual integrity (CI), privacy is shaped and modified by individual, social, and cultural expectations and norms [26]. CI shows that the information flow is governed by context-dependent norms. These norms are characterized by general institutional and social circumstances; the actors involved and their roles; the information being collected, processed, or shared; and the expected transmission principles [32]. The context of the user defines what is appropriate to share. For example, doctors are expected to keep patient information confidential. In an emergency, it would be appropriate to share patient information with other medical professionals, even if explicit consent was not provided. Previous works [2, 33] have used the theory of CI to represent and analyse privacy policies; however, there has been no formal method to represent and reason about the dynamism of contextual norms and to resolve the inconsistencies that arise in privacy policies. Moreover, handling this problem from a multi-party perspective requires new techniques to reason about privacy conflicts.

In this paper, we introduce an agent-based framework for privacy policy reasoning. Our framework is unique in the sense that it combines ontological reasoning with argumentation-based dialogues to make contextual integrity theory practical. For this, we define a new ontology, PROCI, to formally represent the privacy domain.
We first introduce the CI theory. Our proposed framework consists of two main components to implement this: an ontology to formalize CI and an argumentation-based framework to be used by agents for multi-party privacy conflict resolution. We briefly discuss these two components before introducing our framework in Section 4.

3 BACKGROUND

We first introduce the CI theory. Our proposed framework consists of two main components to implement this: an ontology to formalize CI and an argumentation-based framework to be used by agents for multi-party privacy conflict resolution. We briefly discuss these two components before introducing our framework in Section 4.

3.1 Privacy as Contextual Integrity

The theory of contextual integrity (CI) is particularly useful for understanding privacy expectations and the norms of information transmission in a given context [27]. Nissenbaum suggests that information should be distributed and protected according to the norms governing different social contexts.

In CI, four key components are proposed to formalize privacy: (i) Contexts are the situations in which the information flows occur; (ii) Actors include senders of information, recipients of information, and information subject (whom the information is about); (iii) Attributes are defined as the types of information in the information flow; and (iv) Transmission Principles (such as consent, sell, with notice, with a warrant, with authorisation and so on) are the constraints to the information flow from one party to another in a given context. Nissenbaum suggests that contextual integrity is maintained when two types of norms are upheld: Norms of appropriateness and Norms of dissemination. Norms of appropriateness dictate the type of information about an individual that is appropriate to be revealed in a particular context. Norms of dissemination govern the flow of a third party’s personal information from one user to the other.

Representing contexts is a challenging task, and there are various formulations in the literature. Kokciyan and Yolum [18, 20] represent a context as a collection of privacy policies that are semantically similar to each other, where similarity is based on the textual content of the privacy policies. Abowd et al. [1] define context as “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves.” Barth et al. [3] define context as the concept that captures the idea that people act and transact in a society not simply as individuals, but as individuals in certain capacities (roles) in distinctive social contexts, such as healthcare, education, etc.

3.2 A Formal Language to Implement CI

In our approach, we formalize privacy contexts based on CI theory using a formal language defined as an ontology. An ontology is a way to represent knowledge in a specific domain [23]. We use Web Ontology Language (OWL) to model an ontology for privacy contexts. OWL is characterized by its ability to construct rapid data modeling and enable automatic reasoning [9] and it is more expressive than other ontology languages such as RDF [11]. OWL ontologies consist of classes (i.e., domain concepts) and properties (the relationships between these concepts). They describe instances that are individuals belonging to classes. The ontology captures the semantic rules for the norms that govern information flow in a social context. However, we argue that these norms can be subjective, and their justification may need to be explained or argued about to resolve conflicts. For example, assume that an agent tries to represent a norm regarding when a doctor could share sensitive information about a patient with other doctors without the explicit consent of the patient. The representation of this norm, together with its justification, could vary widely based on cultural differences.

An important step towards ensuring contextual integrity is to allow for communication between agents (human or artificial). These agents must communicate to resolve differences and conflicts of opinions or simply inform each other of pertinent facts about contexts and the impact of changing contexts on privacy policies,
3.3 Argumentation Frameworks

In Dung’s abstract argumentation frameworks (AFs) [6], an argument is considered an atomic entity and hence only its interactions with other arguments are modelled. In these frameworks, the internal structure of the arguments and attacks between them is not specified. The emphasis is on the evaluation of the interactions between the arguments to reach a conclusion. [30]. Definition 3.1 provides a formal description of an AF, which is basically a directed graph.

**Definition 3.1. (Dung’s Argumentation Frameworks (AFs))** [6]. \( \mathcal{F} = (\mathcal{A}, \mathcal{D}) \) is an AF if \( \mathcal{A} \) is a set of arguments and \( \mathcal{D} \subseteq \mathcal{A} \times \mathcal{A} \) is a binary defeat relation\(^1\) over \( \mathcal{A} \). Let \( \mathcal{E} \subseteq \mathcal{A} \):

- \( \mathcal{E} \) is conflict-free iff there exists no \( \phi_1, \phi_2 \in \mathcal{E} \) such that \( (\phi_1, \phi_2) \in \mathcal{D} \). Let \( \phi_1 \in \mathcal{E} \) defines \( \phi_1 \) iff for every \( (\phi_2, \phi_1) \in \mathcal{D} \), there exists a \( \phi_3 \in \mathcal{E} \) such that \( (\phi_3, \phi_2) \in \mathcal{D} \). \( \mathcal{E} \) is an admissible set iff \( \mathcal{E} \) is conflict free and defends all its elements. \( \mathcal{E} \) is a complete extension iff there are no other elements which it defends. \( \mathcal{E} \) is a preferred extension iff it is maximal (with respect to set inclusion) complete extension.

AFs offer semantics for the evaluation of arguments to compute extensions. Here, we focus on preferred semantics, which admits multiple extensions. A preferred extension represents a potentially justified view (which may conflict with other views). If an argument is present in all extensions, then it is sceptically justified; while if it is present in at least one extension, it is credulously justified.

3.3.1 Structured Argumentation. We now define the arguments semantically so that the agents could discuss their different points of view. A general framework for giving structure to arguments is the ASPIC+ framework [24, 30]. ASPIC+ defines an argumentation system built from a logical language \( \mathcal{L} \) and defines arguments as inference trees formed by applying strict or defeasible rules to premises that are well-formed formulae (wff) in \( \mathcal{L} \). A strict rule means that if one accepts the antecedents, then one must accept the consequent no matter what. A defeasible rule means that if one accepts all the antecedents, then one must accept the consequent if there is insufficient reason to reject. The notion of attack between arguments means that a certain wff is contrary or contradictory of certain other wff. Definition 3.2 captures a formal definition of an argumentation system.

**Definition 3.2. (Argumentation system).** An argumentation system is a triple \( \mathcal{A}S = (\mathcal{L}, \mathcal{R}, n) \) where (i) \( \mathcal{L} \) is a logical language closed under negation (\( \neg \)), (ii) \( \mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d \) is a set of strict \( (\mathcal{R}_s) \) and defeasible \( (\mathcal{R}_d) \) inference rules of the form \( \phi_1, \ldots, \phi_n \Rightarrow \phi \) and \( \phi_1, \ldots, \phi_n \Rightarrow \phi \) respectively (where \( \phi_1, \phi \) are meta-variables ranging over the wff in \( \mathcal{L} \)), and such that \( \mathcal{R}_s \cap \mathcal{R}_d = \emptyset \). (iii) \( n : \mathcal{R}_d \rightarrow \mathcal{L} \) is a naming convention for defeasible rules. \( n(r) \) is a wff in \( \mathcal{L} \) which says that the defeasible rule \( r \) in \( \mathcal{R} \) is applicable.

Arguments in ASPIC+ are constructed from a knowledge base \( \mathcal{K} \), which contains two disjoint kinds of formulae: axioms \( \mathcal{K}_a \) and ordinary premises \( \mathcal{K}_p \). This distinction is important since the agents could attack arguments derived from assumptions. An argumentation theory is a tuple \( \mathcal{AT} = (\mathcal{A}, \mathcal{K}) \) where \( \mathcal{A} \) is an argumentation system and \( \mathcal{K} \) is a knowledge base in \( \mathcal{A} \). The formal definition of an argument and an attack is provided in Definition 3.3.

**Definition 3.3. (Argument and Attack)** [24]. An argument \( A \) based on an argumentation theory \( (\mathcal{A}, \mathcal{K}) \) with knowledge base \( \mathcal{K} \) and an argumentation system \( (\mathcal{L}, \mathcal{R}, n) \) is: (i) \( \phi \) if \( \phi \in \mathcal{K} \) with: \( \text{Prem}(A) = \{\phi\} \); \( \text{Conc}(A) = \{\phi\} \); \( \text{Sub}(A) = \{\phi\} \); \( \text{TopRule}(A) = \text{undefined} \). (ii) \( A_1, \ldots, A_n \rightarrow \psi \) if \( A_1, \ldots, A_n \) are arguments such that there exists a strict or a defeasible rule \( \text{Conc}(A_1), \ldots, \text{Conc}(A_n) \rightarrow \psi \) in \( \mathcal{R}_s[\mathcal{R}_d] \text{ with } \text{Prem}(A) = \text{Prem}(A_1) \cup \ldots \cup \text{Prem}(A_n), \text{Conc}(A) = \{\psi\}, \text{Sub}(A) = \text{Sub}(A_1) \cup \ldots \cup \text{Sub}(A_n) \cup \{A\} \). (TopRule)(A) = Conc(A), \( A_1, \ldots, A_n \rightarrow \psi \). (iii) \( A \) attacks \( B \) iff \( A \) undercut, rebuts, or undermines \( B \), where \( A \) undercut \( B \) (on \( B' \)) iff \( \text{Conc}(A) = \neg \text{Prem}(r) \) for some \( B' \in \text{Sub}(B) \) such that \( B' \)’s top rule \( r \) is defeasible. \( B \) (on \( B' \)) iff \( \text{Conc}(A) = \neg \text{Prem}(r) \) for some \( B' \in \text{Sub}(B) \) of the form \( B'_1, \ldots, B'_n \Rightarrow \phi \). \( A \) undermines \( B \) (on \( A' \)) iff \( \text{Conc}(A) = \neg \phi \) for an ordinary premise \( \phi \) of \( B \). Note that \( \text{Prem} \) returns all the formulas of \( \mathcal{K} \) called premises used to build the argument, \( \text{Conc} \) returns its conclusion, \( \text{Sub} \) returns all its sub-arguments and \( \text{TopRule} \) returns the last inference rule used in the argument.

ASPIC+ defines a set of arguments with binary relation of defeat, that is, it defines argumentation frameworks in the sense of AFs [6], thus making the semantics of abstract argumentation applicable to ASPIC+ (Definition 3.1).

![Figure 1: Agent-Based Framework for Privacy Policy Reasoning](image)

**Figure 1:** Agent-Based Framework for Privacy Policy Reasoning, where agent reasoning is based on the PROCI ontology and the argumentation-based dialogue system.

### 4 AGENT-BASED FRAMEWORK FOR PRIVACY POLICY REASONING

In this section, we describe our proposed framework consisting of two main components: (i) A privacy ontology based on contextual integrity (PROCI), which formally specifies the rules that agents

---

\(^1\)A defeat relation is a version of the attack relation indicating a successful attack relation.
must take into account when taking on various roles in contexts and requesting access to (or granting access to) information containing attributes of the users they represent. (ii) An argumentation-based dialogue framework that addresses communication between agents and reasoning about privacy policies. The framework uses the syntax and semantics of the argumentation frameworks presented in Section 3.3 to represent and evaluate privacy dialogues.

Figure 1 depicts our agent-based framework for privacy policy reasoning. The left part of the figure depicts the workflow of a privacy ontology that aims to generate privacy decisions based on some specified norms. The argumentation-based dialogue system manages the different types of dialogues that are possible to initialize. This framework can be instantiated as an interactive dialogue process in each instance of a recommendation for a dialogue.

### 4.1 Privacy Ontology based on Contextual Integrity (PROCI)

We develop an ontology called PROCI\(^2\) for contextual integrity. The ontology consists of six distinctive but related classes as depicted in Figure 2: (i) Agent; represents a set of agents (i.e., both human and software agents) and their associated instances and attributes, (ii) Role; represents a set of all roles that are presently associated with an agent, (iii) Context; represents a set of disjoint social contexts (e.g., professional and emergency contexts) that describe the agents, (iv) Information; represents a set of facts received or learned about an agent (e.g., an agent's location or educational qualification, etc.) (v) Situation; represents a set of situational conditions or activities of an agent (e.g., an emergency situation, attending a class, presenting a seminar, etc.), and (vi) Organisation; represents a set of social or professional organisations that an agent is part of (e.g., a professional association, a religious group, a company, etc.). The ontology also defines a set of properties and relationships that are associated with these classes. In figure 2, note that solid lines represent properties while dashed lines represent subclasses. Following the theory of contextual integrity (CI) as discussed in Section 3.1, PROCI includes the following components:

#### 4.1.1 Contexts, Actors, Roles

Classes such as PRContext, FRContext, and EMContext are used to describe social contexts (professional, friendship, and emergency) of online users, and each of them is a subclass of Context. Within a social context, the class Agent describes the actors that are involved. We make a distinction between human and software agents since dialogues could be initiated with human agents as well. This class has four subclasses: User; is a set of human agents (natural person) who own information or who can be identified directly or indirectly by the information, ThirdParty; is a set of human agents that request access to user information, UserAgent; the class of software agents that represent users, and ThirdPartyAgent; the class of software agents represent third parties. Each Agent has a role in a specific context; and some instances are modelled as :admin-officer, :line-manager, :em-responder :health-and-safety-manager. To connect agents with roles in a social context, hasRole links Agents to Roles. We define properties such as isColleagueOf and isFriendOf to represent the relationships between agents. The following example represents a privacy scenario in an OSN that could be represented by the instantiation of PROCI.

**Example 1.** Alice, Bob, and Charlie are all colleagues of each other. Bob prefers to share his location information with Alice and Charlie in a professional context and does not want to reveal this information in a friendship context. Alice is the emergency responder and also the health and safety manager; whereas Charlie is the administrative officer.

Assume that we model a personal privacy assistant for Bob in this example. An instance of User will be instantiated as :bob, while :alice and :charlie are instances of ThirdParty. Moreover, user/third party will also have an instance for their software agents. :bob-a is an instance of UserAgent while :alice-a and :charlie-a are instances of ThirdPartyAgent, respectively. Bob owns some pieces of information such as :bob-loc and :bob-mobile. Other agents can make AccessRequests to access to such data, bob-a will be the agent making privacy decisions.

An Agent can be in a context with other agents, inContext is used to specify this; and the agents can be in multiple contexts [5, 20]. For example, :bob, :alice and :charlie can be in a PRContext and an EMContext at the same time, and in such combined contexts, it may be appropriate for :alice with the role :health-and-safety-manager to have access to :bob’s mobile information (:bob-mobile) but not :charlie with the role :admin-officer.

#### 4.1.2 Norms of Appropriateness/Dissemination

We define three types of norms to govern the information flow in a social context, as introduced in Section 3.1: Access norms (A), Inference norms (I) and Privacy norms (P). The access norms generalize the norms of appropriateness and norms of dissemination of contextual integrity. These norms specify whether access requests are allowed or denied; the allow and disallow predicates appear in the head of access norms. An access norm A allows a third party to access private

---

\(^2\)The ontology is shared as part of our supplementary material. Note that PROCI is populated with a representative list of classes and relations.
information of a user if its conditions are satisfied; otherwise, the access is denied. Inference norms enable a software agent (a user agent or a third-party agent) to derive new information from existing knowledge in its ontology. In a privacy norm, the user specifies which type of access request should be granted or rejected.

In this work, we assume that the software agents are aware of the norms of their users. Some related work focuses on learning such norms [16, 21] which is out of our scope. Semantic Web Rule Language (SWRL) [10] is used to represent the CI norms. The norms are defined as a set of preconditions and consequent states consisting of the conjunction of atoms (i.e., Body → Head), which means that if the body holds, then the head must hold. Here, the atoms are of the form \( c(x) \) and \( P(x,y) \) where \( c \) is a class name (e.g., Information) and \( P \) is a property name (e.g., hasRole), which are defined in the ontology. \( x \) and \( y \) are variables prefixed with a question mark (e.g. \(?x\) or \(?y\)).

In Table 1, we show an example set of norms of inference (1), privacy preference (9) and access request (4). Each norm is modeled as a SWRL rule. For example, Bob’s agent uses the three inference norms \( I_1, I_2, \) and \( I_3 \) to infer whether a user and a third party are in a friendship, professional, or emergency context, respectively. For example, the norm \( I_3 \) states that if there is an emergency situation and a user (?user) is in the emergency situation, and there is a third party (?tp) with a role of emergency responder, then ?user and ?tp are in an emergency context. Bob has two privacy norms \( P_1 \) and \( P_2 \) that state the contexts in which he prefers to share (resp., not share) his location information with a third party. Access norms \( A_1, A_2 \) and \( A_3 \) are used by Bob’s agent :bob-a to allow or deny access to Bob’s location information for access requests from Alice (a third party). For instance, norm \( A_3 \) states that if there is an access request from a third party :alice for Bob’s location information :bob-loc (that is, hasAR :alice, bob-loc), the access request should be denied in a friendship context inFRContext:bob, alice). In the protocol, the following moves are defined: assert(Agi, φi, t), contradict(Agi, φi, φj, t), retract(Agi, φi, t), exit and update. Note that \( Agi \) represents the user agent or a user/third-party agent who made a move. A user agent uses the assert(Agi, φi, t) move context in which the user can share a private information (reasoning about contexts). Similarly, a user agent may need to negotiate with a third party agent to resolve multiparty privacy conflict (reasoning about MPPCs). Here, argumentation-based dialogues are conceived as the underlying mechanism by which an agent communicates with a specific user or with a third party to resolve privacy conflicts.

### 4.2.1 The Dialogue Protocol

A dialogue between a user agent and a specific user or between a user agent and a third party agent consists of a sequence of moves, where each move references both a statement and the user agent/user who made the statement. A statement can be a request for information, a provided information, or a privacy decision to grant (resp., decline) an access request. This is captured in Definition 4.2.

**Definition 4.2.** A dialogue \( D \) consists of a sequence of iterations such that \( D = \left[ \left( M_1, \ldots, M_n \right), \ldots, \left( M_n, \ldots, M_1 \right) \right] \), which involves \( n \) participants \( p_1, \ldots, p_n \) where \( n \geq 2 \). Within a dialogue \( D \), iteration \( j \) consists of a sequence of moves \( \left( M_1, \ldots, M_n \right) \).

A user agent evaluates the set of statements exchanged within an iteration to update its privacy knowledge. Within each iteration, there is a claim to be discussed and arguments that attack or defend the claim. Note that a claim is abstractly represented as an argument. An iteration therefore represents a sub-discussion focused around a single topic of the dialogue, which can be treated in an atomic manner.

The dialogue protocol is depicted in Figure 3, which describes the set of legal moves that are permitted in each iteration. Each node represents the type of move an agent can make in a dialogue, and the outgoing arcs of a node indicate possible responding moves. In the protocol, the following moves are defined: assert(Agi, φi, t), contradict(Agi, φi, φj, t), retract(Agi, φi, t), exit and update. Note that \( Agi \) represents the user agent or a user/third-party agent who made a move. A user agent uses the assert(Agi, φi, t) move

### 4.2 Argumentation-based Dialogue Framework

There are instances where a user agent needs to negotiate with a specific user about whether or not to share a private information of the user should be shared (reasoning about user privacy preferences) or the...

<table>
<thead>
<tr>
<th>Table 1: Some Norms for Bob as SWRL Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1: \text{isFriendOf}(?user, ?tp) \rightarrow \text{inFRContext}(?user, ?tp) )</td>
</tr>
<tr>
<td>( I_2: \text{workAt}(?user, ?office), \text{workAt}(?tp, ?office), \text{hasRole}(?tp, \text{role}), \text{isColleagueOf}(?user, ?tp) \rightarrow \text{inFRContext}(?user, ?tp) )</td>
</tr>
<tr>
<td>( I_3: \text{Emergency}(?km), \text{isInEmergencyOf}(?user, ?km), \text{hasRole}(?tp, \text{em-responder}) \rightarrow \text{inEMContext}(?user, ?tp) )</td>
</tr>
<tr>
<td>( P_1: \text{owns}(bob, bob-loc), \text{inFRContext}(bob, ?tp) \rightarrow \text{allow}(?tp, bob-loc) )</td>
</tr>
<tr>
<td>( P_2: \text{owns}(bob, bob-loc), \text{inFRContext}(bob, ?tp) \rightarrow \text{disallow}(?tp, bob-loc) )</td>
</tr>
<tr>
<td>( A_1: \text{owns}(bob, bob-loc), \text{inFRContext}(bob, alice), \text{hasAR}(alice, bob-loc) \rightarrow \text{allow}(alice, bob-loc) )</td>
</tr>
<tr>
<td>( A_2: \text{owns}(bob, bob-mobile), \text{inEMContext}(bob, alice), \text{hasAR}(alice, bob-mobile) \rightarrow \text{allow}(alice, bob-mobile) )</td>
</tr>
<tr>
<td>( A_3: \text{owns}(bob, bob-loc), \text{inFRContext}(bob, alice), \text{hasAR}(alice, bob-loc) \rightarrow \text{disallow}(alice, bob-loc) )</td>
</tr>
</tbody>
</table>
to start a dialogue by submitting a claim $\phi$ in iteration $t$. The claim could be a context inferred from its ontology that the user it represents or a third party agent needs to be aware of (reasoning about context) or an explanation of a privacy decision in a case where the privacy decision is in conflict with the privacy preference of the user or the third party agent (reasoning about MPPC) or a new information for the user to update its privacy preference (reasoning about user privacy preferences). Similarly, a user agent uses the $\text{contradict}(Ag_i, \phi_1, \phi_2, t)$ move labeled $M^i_{t+1}$ to make a claim $\phi_2$ to attack a previous claim $\phi_1$ made by a user/third party agent in move $M^i_t$. The claim $\phi_2$ of the move $M^i_{t+1}$ must be relevant to the claim $\phi_1$ of the move $M^i_t$. Specifically, a claim $\phi_2$ is relevant to $\phi_1$ if it attacks it (cf. Definition 3.1). A user agent or a user/third party agent uses the $\text{retract}(Ag_i, \phi, t)$ move to retract its previous claim. A user agent or a user/third party agent uses the $\text{exit}$ move to exit an iteration. This move is made when the user agent or a user/third party agent has no more $\text{contradict}(Ag_i, \phi_1, \phi_2, t)$ move to advance within the iteration. When an iteration is completed (shown by the terminal node $\text{update}$ in the figure), the dialogue state is updated. The dialogue then proceeds to the next iteration or may terminate.

4.2.2 Combining PROCI with Dialogues. We assume that a user agent or a user/third party agent usually has a set of norms/rules to generate its set of arguments in a dialogue. At any point in time, a user agent or a user/third party agent may add argument to a dialogue if it is not already present within it or retract argument (if it was already present) by adding another argument that indicates retraction of a previous argument. Therefore, a dialogue usually results in a set of arguments and attack or defeat relations arising from conflicts between arguments. In our framework (Figure 1), we apply the ASPIC+ argumentation theory to represent and evaluate a dialogue. Note that we do not consider an ordering $\preceq$ of the elements of $\mathcal{K}$ [24]. To resolve an attack between arguments to a defeat, an attacker wins unless the attacked argument is defended by other undefeated arguments. We define the mapping from PROCI to our framework in Definition 4.3.

Definition 4.3. Given PROCI and $\mathcal{AS} = (\mathcal{L}, \mathcal{R}, n)$, with knowledge base $\mathcal{K}$, in our framework, we adapt ASPIC+ argumentation theory $\mathcal{AT} = (\mathcal{AS}, \mathcal{K})$ as follows:

1. $\mathcal{L}$ is the formal language defined in PROCI; arguments and attacks are defined based on Definition 3.3; contrariness is as defined in Section 3.3. For simplicity, we assume that different contexts are contrary to each other.
2. $\mathcal{K} = \mathcal{K}_p \cup \mathcal{K}_n$, where $\mathcal{K}_p = \{ e \in \mathcal{L} | e$ is a property/class axiom in PROCI $\}$ such that: (i) In PROCI, $\mathcal{K}_p$ is a set of properties/property axioms. For example, the property $\text{hasRole}(alice, \text{admin-officer})$ is an ordinary premise that can be attacked. (ii) $\mathcal{K}_n$ is a set of class/class expression axioms in PROCI. For example, $\text{disjointClasses}(\text{User}, \text{ThirdParty})$ is a class axiom from PROCI. Intuitively, arguments cannot be attacked on their axiom premises.
3. $\mathcal{R} = \mathcal{R}_d \cup \mathcal{R}_e$ is the smallest set of inference, privacy, and access rules such that: (i) Inference rules are strict rules set by the user agent as default rules; hence, they are mapped to $\mathcal{R}_d$. (ii) Access and Privacy rules in the ontology are subjective perceptions and/or evaluations of the user agent and user/third party agent, respectively, and are therefore mapped to defeasible rules ($\mathcal{R}_e$).

A user agent $Ag_i$ evaluates the arguments that have been put forward in a dialogue. Definition 4.4 formally defines an agent-specific argumentation framework.

Definition 4.4 (Agent-specific Argumentation Framework). Let $Ag_i$ be a user agent, the argumentation framework of $Ag_i$ is a tuple $\mathcal{AF}_{Ag_i} = (\mathcal{A}_D, \mathcal{D}_D)$ where $\mathcal{A}_D$ is a set of arguments in a dialogue $D$ and $\mathcal{D}_D \subseteq \mathcal{A}_D \times \mathcal{A}_D$ is a defeat relation.

Note that an $\mathcal{AF}_{Ag_i}$ can have multiple extensions when the user agent performs the reasoning under preferred semantics [6]. To define the set of justified arguments in our framework, we borrow the notions of sceptical and credulous acceptability (Definition 4.5).

Definition 4.5 (Acceptability). Given $\mathcal{AF}_{Ag_i} = (\mathcal{A}_D, \mathcal{D}_D)$, a set of arguments $\mathcal{E}_D \subseteq \mathcal{A}_D$ in $\mathcal{AF}_{Ag_i}$ is a preferred extension for a user agent $Ag_i$ if it is a maximal (with respect to set inclusion) complete extension obtained from $\mathcal{AF}_{Ag_i}$. An argument $\phi$ is sceptically justified if it is in every $\mathcal{E}_D$ obtained from $\mathcal{AF}_{Ag_i}$, and the other hand, $\phi$ is credulously justified if it is in some $\mathcal{E}_D$.

4.3 Dialogue between a User Agent and a User about Privacy Preferences and Contexts

In Table 2, we illustrate a dialogue between a user (Bob) and his agent. The example shows the conflict between Bob’s preference and the privacy decision of his agent regarding a third-party’s request for his location information. This example also shows how a dialogue could help agents update their information about privacy contexts. Consider the following example where Alice wants to access Bob’s location information.

**Example 2.** There is a gas explosion reported near Bob’s residence. Alice is worried about Bob and wants to access his location. This access request results in a dialogue between Bob and his agent bob-a.

We use the following notation to represent the moves of the agents in a dialogue. In a move $M^y$, $x$ denotes the move index.
number, while $y$ denotes the iteration in which the move was made. In Table 2, the move $M^1_1$ is an assert($\theta$, $\phi$, $t$) move of Bob’s agent (bob-a) to start the dialogue. The argument $\phi$ in the move is generated from the rules inferred from the agent ontology, and suggests that the location information should be shared. The move $M^2_1$ is a contradict($\theta$, $\phi$, $\psi$, $t$) move of Bob, because Bob does not want to share his location information in a friendship context (i.e., on Facebook). bob-a then makes a move to explain to Bob that there was an explosion nearby and therefore the users were in an emergency context. In other words, bob-a makes the move $M^2_1$ to contradict($\theta$, $\phi$, $\psi$, $t$). The argument $\neg \phi$ in move $M^2_1$ by Bob can be generated from the rules that are inferred from its privacy settings on OSNs and/or can be taken as input, as we consider here. We map these rules to the adapted ASPIC+ argumentation theory (see Table 4.3). Note that the argument in move $M^1_1$ is rebutted by the argument in move $M^2_1$. Furthermore, the rules in moves $M^1_1$ and $M^2_1$ are the access and privacy rules of Bob’s agent and Bob (that are involved in the dialogue, respectively) and are represented as defeasible rules (c.f., Definition 4.3). The argument in move $M^2_1$ defeats the argument in $M^1_1$, as it is an undefeated argument. The inference rule in the move $M^1_1$ is represented as a strict rule (c.f., Definition 4.3). Since Bob did not make any further contradict($\theta$, $\phi$, $\psi$, $t$) move, Bob’s agent $A\theta$ made an exit move to exit the iteration and an update move to update the dialogue state, respectively. Note that we do not represent these two moves in Table 2 as their representation is trivial.

Table 2: An example dialogue between an agent and its user to agree on a privacy decision

<table>
<thead>
<tr>
<th>Moves/Players</th>
<th>SWRL Rules</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M^1_1$ by :bob-a :alice would like to access your location on Facebook</td>
<td>$X_1$: owns(bob, bob-loc)</td>
<td>$\phi$: share location with :alice</td>
</tr>
<tr>
<td>$M^2_1$ by :bob Bob wants to apply his privacy setting in a friendship context</td>
<td>$X_1$: owns(bob, bob-loc)</td>
<td>$\neg \phi$: do not share location with :alice</td>
</tr>
<tr>
<td>$M^3_1$ by :bob-a A gas explosion has been reported near your residence</td>
<td>$Y_1$: Emergency(gas-exp)</td>
<td>$\exists \gamma$: there is an emergency</td>
</tr>
</tbody>
</table>

Arguments in Table 2 and their corresponding defeat relations are abstracted into an ASPIC+ argument graph, as shown in Figure 4a where the rectangle with a solid line denotes an axiom {$Y_1$}, while those with dotted lines denote ordinary premises such as $X_1$ and $Y_2$. Strict rules are denoted with solid lines, whereas defeasible rules are denoted with dotted lines. Rules $A\theta$, $P_\theta$ and $I_\theta$ denote the access, privacy, and inference rules, respectively. $X_{AI}$ denotes an argument moved by Bob’s agent, while $X_{UJ}$ denotes the one moved by Bob. An argument $X_{AI}$ for $\phi$ (i.e. with conclusion $\phi$) is shown in Figure 4a with the premises at the bottom and the conclusion at the top of the tree. The directed arrows denote defeat relations. The ASPIC+ graph can be further abstracted into the AF (Figure 4b). The justified arguments of AF are then evaluated using the preferred semantics of AFs as $[\phi, \alpha]$ [6]. In our example, Alice is granted access to Bob’s location information :bob-loc as arguments $\phi$ and $\alpha$ are skeptically justified.

Figure 4: (a) depicts the dialogue between two agents (Table 2). (b) is the abstract argument graph used for the evaluation of the winning arguments under preferred semantics.

4.4 Handling Multi-Party Privacy Conflicts

In this section, we focus on a multi-party dialogue between two agents where a third-party agent (:charlie-a) dialogue with Bob’s agent about whether or not to share Bob’s data with another third-party (:alice-a). Hence, the agents try to agree on a privacy decision automatically by conducting a dialogue. Note that the complete set of rules can be found in the PROCI ontology. Consider the following example where multi-party privacy conflicts occur.

Example 3. Bob is involved in an accident. alice-a, the agent of Alice, is asking the agent of Charlie (charlie-a) to share Bob’s salary information. charlie-a is trying to persuade the agent of Bob (bob-a) to share Bob’s salary information with alice-a.

Charlie may have access to Bob’s salary information (as an admin-officer), while Alice is denied such access by the agent of Bob as she is the health-and-safety-manager. The norms of dissemination prevent Charlie from sharing Bob’s salary information with Alice without Bob’s consent. We represent a sample dialogue in Table 3. The argument of move $M^1_1$ by :charlie-a claims that Bob’s salary information should be shared with a third-party (Alice), because charlie-a has information about Bob being involved in an accident. This piece of information is missing for bob-a at the time of the dialogue; hence, this argument is attacked by the argument in move $M^2_1$ by Bob’s agent :bob-a based on the access rules defined in its ontology. The argument of move $M^1_1$ by :charlie-a defeats the argument of move $M^2_1$. Now that bob-a has information about the accident context, it is appropriate to share Bob’s salary information.
Table 3: An example dialogue between two agents (:bob-a and charlie-a) that involves three different users.

<table>
<thead>
<tr>
<th>Moves/Players</th>
<th>SWRL Rules</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_1 ) by :charlie-a Alice</td>
<td>( X_c: owns(bob, bob-salary) )</td>
<td>( \psi: \text{share} )</td>
</tr>
<tr>
<td>should access Bob’s salary</td>
<td>( X_c: hasRole(alice, h-safety-officer) )</td>
<td></td>
</tr>
<tr>
<td>information</td>
<td>( X_c: inAccContext(bob, alice) )</td>
<td>Bob’s salary</td>
</tr>
<tr>
<td></td>
<td>( X_c: hasAR(alice, :bob-salary) )</td>
<td>information</td>
</tr>
<tr>
<td></td>
<td>( \Rightarrow \psi: allow(alice, :bob-salary) )</td>
<td>with Alice</td>
</tr>
<tr>
<td></td>
<td>( M_2 ) by :bob-a Alice should</td>
<td>( \neg \psi: \text{do not} )</td>
</tr>
<tr>
<td>not access</td>
<td>( X_b: hasRole(bob, h-safety-officer) )</td>
<td>share</td>
</tr>
<tr>
<td>Bob’s salary information</td>
<td>( X_b: inPRContext(bob, alice) )</td>
<td>Bob’s salary</td>
</tr>
<tr>
<td></td>
<td>( X_b: hasAR(alice, :bob-salary) )</td>
<td>information</td>
</tr>
<tr>
<td></td>
<td>( \Rightarrow \neg \psi: \text{disallow(alice, :bob-salary)} )</td>
<td>with Alice</td>
</tr>
<tr>
<td></td>
<td>( M_3 ) by :charlie-a The information</td>
<td>( \phi: \text{Bob was} )</td>
</tr>
<tr>
<td>is needed to compensate for the injury</td>
<td>( X_c: hasRole(alice, h-safety-officer) )</td>
<td>involved</td>
</tr>
<tr>
<td></td>
<td>( Y_i: inAccident(bob, true) )</td>
<td>in accident</td>
</tr>
</tbody>
</table>

with Alice, and Charlie sharing this information with Alice would not violate the norms of dissemination anymore.

The example in Section 4.3 illustrates how an argumentation-based dialogue can be used as a mechanism to ensure appropriate access to the information of a user (i.e., norms of appropriateness of CI); here, we illustrate the norms of dissemination. It is important to note that this type of dialogue is useful to facilitate exchange of explanations between agents/users over privacy preferences and expectations. The explanations are important to improve the understanding of users about particular privacy situations. We will work on the explanation aspects of our framework as part of our future work. Furthermore, some agents may be malicious or incompetent, and — to achieve desirable dialectical outcomes — the inputs from these agents should be discounted for lack of trust [28].

4.5 Theoretical Results

We have implemented the PROCI ontology using the Protégé tool\(^3\), which has automated reasoners (e.g., Pellet) to detect inconsistencies and redundant knowledge. The consistency of the ontology depends on the reasoner being used to apply further inference. However, due to the overlap of social contexts or the hierarchical relationship between actors, their roles, and attributes, it is expected to come up with inconsistencies in the formulation of rules for a privacy policy. For example, there may be rules for both positive and negative authorisation of access to the same piece of information, resulting in privacy conflicts; which should be addressed externally to increase transparency in the reasoning process as we do here by using argumentation. Definition 4.6 defines what a consistent ontology instance is.

\(^3\)http://protege.stanford.edu/

Definition 4.6 (Consistency). A PROCI instance \( \theta \) is consistent for a privacy decision \( \alpha \) if there is no contradiction in the set of access rules \( A_1, \ldots, A_n \) and privacy rules \( P_1, \ldots, P_m \) in PROCI or the set of arguments \( X_1, \ldots, X_n \) in \( AT \) that support \( \alpha \). That is, \( A_1, \ldots, A_n \land P_1, \ldots, P_m \mid X_1, \ldots, X_n \Rightarrow \alpha \).

**Theorem 4.7.** Inference, privacy, and access rules are consistent in the ontology given a particular privacy scenario.

**Proof Sketch.** Each agent is equipped with an ontology that is consistent thanks to the underlying reasoner, which is Pellet [34] in our case. Pellet is sound and complete with respect to OWL. Therefore, each of the rules in the ontology is semantically consistent. However, domain-specific conflicts may arise between different rules, which we address through argumentation-based dialogues. To resolve inconsistencies between conflicting rules, we apply the semantics of the ASPIC+ and Dung argumentation frameworks (see Section 4.3 and Tables 2 and 3).

**Theorem 4.8.** For any \( AF_{\text{Ag}} \), where the agent adopts preferred semantics to evaluate arguments, the set of computed extensions (i.e., the sets of justified arguments) are consistent, complete, and sound. The proposed agent-specific argumentation framework will preserve consistency of a PROCI instance to make privacy decisions.

**Proof Sketch.** The preferred semantics ensure that there is an existence of at least one extension. The extensions computed under preferred semantics are conflict-free; therefore, we can say that preferred semantics can be applied to resolve inconsistencies in the rules to obtain a consistent PROCI instance considering the agent’s preferences to choose among alternative extensions. With respect to completeness and soundness, completeness can be proven for a finite set of arguments in a dialogue. The dialogue protocol described in Section 4.2 defines the dialogue moves allowed of the agents in a dialogue, and the criteria for the termination and evaluation of a dialogue. These protocols ensure that a dialogue results in a finite set of arguments that are automatically derived from the agents’ ontology. Therefore, all possible arguments are generated for a particular privacy scenario. Then completeness follows, since every preferred extension is maximal (with respect to set inclusion) complete extension. Likewise, soundness follows, since each of the preferred extensions includes valid arguments derived from the agent’s ontology. Hence, each preferred extension consists of a set of admissible arguments to make a privacy decision.

5 CONCLUSION

In this paper, we have proposed an agent-based framework to handle privacy policy reasoning in social contexts. Our framework builds on the theory of contextual integrity to represent social contexts, the actors involved, and the norms to govern information flow. We have formally defined a new ontology, PROCI, and the argumentation-based dialogue framework. We have presented two examples, where we illustrate how user agent and user/third-party agents apply the inference, privacy, and access rules within their ontologies for privacy policies and how argumentation provides an underlying mechanism for resolving the conflicts. Our theoretical results also provide completeness and soundness guarantees.

As future work, it would be interesting to incorporate machine learning (e.g., reinforcement learning) to learn access rules based on the privacy behaviour of the user in online systems such as OSN and IoT [19]. We also plan to develop a chatbot to manage the dialogue model of our framework, similar to the work done in [14], where the agent could explain the decision-making process to the user. Such an interaction could then be used to choose among alternative privacy decisions while keeping the human in the loop.
ACKNOWLEDGMENTS
This research was funded by the UKRI Strategic Priorities Fund via the REPHRAIN research centre.

REFERENCES
[31] Ramon Ruiz-Dolz, José Alemay, Stella Heras, and Ana Garcia-Fornes. 2019. Automatic Generation of Explanations to Prevent Privacy Violations. In XAILA' JURIX.