PriGuardTool: A Tool for Monitoring Privacy Violations in Online Social Networks

(Demonstration)

Nadin Kökciyan
Department of Computer Engineering
Bogazici University, 34342 Bebek
Istanbul, Turkey
nadin.kokciyan@boun.edu.tr

Pınar Yolum
Department of Computer Engineering
Bogazici University, 34342 Bebek
Istanbul, Turkey
pinar.yolum@boun.edu.tr

ABSTRACT

In this demonstration, we present PriGuardTool, which is a Web-based tool that can detect privacy violations in online social networks and notify the users accordingly. Our tool comes up with an interface where the users input their privacy concerns. An agent represents a user in the online social network. Each agent is responsible for generating commitments between its user and the system to monitor the social network and check for privacy violations. We demonstrate PriGuardTool by using various real-life scenarios.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Distributed Artificial Intelligence *Multiagent systems*

Keywords

Privacy, Online Social Networks, Commitment, Ontology

1. INTRODUCTION

In online social networks, privacy violations can take place in numerous ways: A user herself may misconfigure the system and reveal unintended content; or a friend of a user can share a content not knowing that the user would not want the content online; or sharing of certain information either by the user or others could lead to other information being unleashed unexpectedly. In all cases, the users seek tools that will help them to preserve their privacy and catch privacy breaches if any, so that they can take an action.

We propose PriGuard model that focuses on: (i) specifying the social network, privacy concerns and privacy violations, (ii) detecting privacy violations in the system, and (iii) notifying the user to take an action. PriGuardTool is a Web-based tool that implements PriGuard model. In this demonstration, we will show how PriGuardTool can deal with various scenarios from the literature.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

2. PRIGUARD APPROACH

PriGuard is a commitment-based model for privacy-aware online social networks. Each user in the social network is represented by an agent. We use commitments [2] to capture the privacy promises done by the social network operator to each user, based on the user's specified privacy expectations. Consider the following example where an online social network user Dennis has requested his location to be hidden from his posts. Later, he shares a picture with friends without knowing that the picture includes a geotag. The geotag includes geographical information that will give away his location. Dennis' privacy agreement with the online social network can be represented with the following commitment: C_1(`:osn`; `:dennis`; *isFriendOf*(`:dennis`,X), *isAbout*(P,`:dennis`), `LocationPost`(P); not(*canSeePost*(X,P))). That is, `:osn` commits to `:dennis` to not show his location posts to the user X if `:dennis` declares X to be a friend.

In PriGuard, the social network domain is formally defined using Description Logics (DL). This domain consists of concepts (e.g.; `Agent`), roles (e.g.; *isFriendOf*) and individuals names (e.g.; `:dennis`). Hence, the relationships and the posts can be semantically described by the use of DL as well. The social network operator operates according to norms (i.e.; semantic rules). In PriGuard, norms are specified as Datalog rules. *sharesPost*(X,P) → *canSeePost*(X,P) is a Datalog rule, which states that an agent can see the posts that it shares.

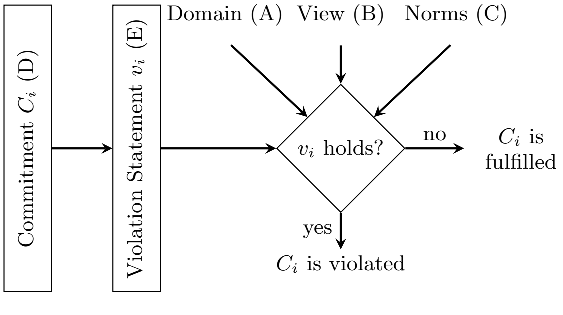
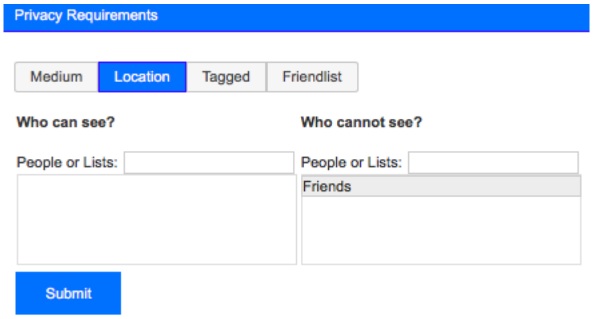


Figure 1: Detection in PriGuard

Detection of Privacy Violations: Figure 1 shows how PriGuard approach works. Recall that there exists a commitment from the social network operator to the user agent to preserve certain privacy conditions. A commitment violation signals a breach of privacy. Hence, the agent computes



Location

Who cannot see?

Friends

Figure 2: Dennis declaring his friends to not see his location posts.

under which conditions a commitment would be violated [2]. Then, the agent uses the domain information, norms, the view information and the violation statement for detecting privacy violations. A view consists of three sets: a set of agents, a set of relationships and a set of posts. An exact view representation of the social network is the global view of the system. However, an agent can choose to focus on subviews of the exact view. Finally, the agent reports whether there is a privacy violation by checking if a given commitment is violated or not. It depends on the creditor of the commitment (e.g., the user) to take an action accordingly.

3. PRIGUARDTOOL

PriGuardTool is a Web-based tool that implements PriGuard model. Each component in Figure 1 is implemented in PriGuardTool. We use ontologies to capture the domain,, view, and norms of the social network.

Domain (A): The social network domain is represented using PriGuard ontology specified in OWL 2 Web Ontology Language [1]. PriGuard model is a DL model, which can be completely defined in an OWL 2 ontology.

View (B): In PriGuard ontology, a view is a set of class assertions (e.g.; ClassAssertion(`Agent :alice`)) and object property assertions (e.g.; ObjectPropertyAssertion (*isFriendOf* `:dennis :charlie`)).

DL Rules (C): In PriGuard, norms are defined as Datalog rules. OWL 2 is an expressive language to represent some Datalog rules as DL rules. For example, consider this rule: $\texttt{Post} \sqcap \exists \mathit{hasMedium}.\exists \mathit{hasGeotag}.\texttt{Location} \sqsubseteq \texttt{LocationPost}$. This rule states that a post that includes a geotagged picture is an instance of `LocationPost` class in the ontology.

Commitments (D): Users input their privacy concerns via PriGuardTool interface as depicted in Figure 2. The user can specify her privacy concerns regarding medium posts, location posts and posts that the user is tagged in. Moreover, the user can declare who can access her friendlist in the social network. For each category, the user declares two groups of people: one group that can see that category and a group that cannot. If the user specifies conflicting privacy concerns (e.g.; a user is part of both groups), the agent adopts a conservative approach to minimize privacy violations to occur; i.e., it finds conflicting users and move them to the group that cannot see the content.

Violation Statements (E): After all the semantic inferences are made by the use of PriGuard ontology and DL rules, the agent should be able to query this knowledge to monitor privacy violations in the social network. For this, we use SPARQL for querying RDF-based information. Note that ontological axioms can also be seen as RDF triples. In a SPARQL query, there are query variables to retrieve the desired results. We only focus on *SELECT* queries with filter expressions *NOT EXISTS* and *EXISTS* to represent violation statements. The violation statement of C_1 is shown as a SPARQL query in Table 1. The abbreviation P declares a namespace prefix. *osn* prefix shows where to find PriGuard ontology for querying. This *SELECT* query declares two query variables (?x and ?p) to be retrieved. The core part of the query is defined in the *WHERE* block, which consists of four triples (one is used in a filter expression). This query returns friends of Dennis who can see his location posts.

Table 1: The Violation Statement of C_1

```
P rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
P osn: <http://mas.cmpe.boun.edu.tr/ontologies/osn#>
SELECT ?x ?p WHERE {
        ?x osn:isFriendOf osn:dennis .
        ?p osn:isAbout osn:dennis .
        ?p rdf:type osn:LocationPost .
        FILTER EXISTS {?x osn:canSeePost ?p} }
```

4. DEMO DETAILS

- We will use four real-life scenarios from the literature to demonstrate how PriGuard approach works in PriGuardTool. Each scenario will address a different type of privacy violation, including inference-based and co-privacy violations.
- We will show how a user can use PriGuardTool interface to input her privacy concerns and how the commitments are generated automatically by ensuring that inconsistencies do not arise.
- For each commitment in the system, we will show how the corresponding violation statements are created.
- For each scenario, we will show how the user will check for privacy violations, and how her agent will report the detection results. We will also interpret these results with comparison to existing work.

Our demo video is available online at: `https://youtu.be/9UJ02h-ud00`.

Acknowledgments

This work is supported by TUBITAK under grant 113E543.

REFERENCES

[1] B. Motik, P. F. Patel-Schneider, B. Parsia, C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, et al. Owl 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation*, 27(65):159, 2009.

[2] P. Yolum and M. P. Singh. Flexible protocol specification and execution: applying event calculus planning using commitments. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 527–534, 2002.