ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1608

Strategies for Privacy Negotiation in Online Social Networks

Dilara Kekülllüoğlu[1] and **Nadin Kökciyan**[2] and **Pınar Yolum**[3]

1 Introduction

Online social networks are changing the way information is shared among individuals. Contrary to traditional Web systems, such as e-commerce Web sites, where information about a user is managed solely by the user herself, in online social networks other users can contribute to the content that is shared about an individual. The shared content may reveal information about the user, which the user might not wanted to share herself. This creates a privacy breach on the user's side. In current online social networks, a common way to deal with this is for the user to complain to the social network administration and ask the content to be removed. However, by the time the content is removed (if at all), many people might have seen it already. Ideally, it would be best if such a content was not shared in the first place.

Recent work on privacy management has focused on applying agreement technologies to solve privacy problems before they take place. Two important works in this line are that of Mester *et al.* [3] and Such and Rovatsos [4]. These approaches apply negotiation techniques [2] to resolve privacy conflicts among users. They both consider negotiation before a content is being shared. Both approaches assume that negotiation is being performed on a single content and cannot account for ongoing interactions. However, it has been observed that users build reciprocal trust in online social networks and respect others as much as others respect them. Hence, it is of utmost importance to consider repeated interactions, as opposed to single interactions, to study privacy leakages.

This paper proposes a multiagent management of privacy in online social networks, where each user is represented by an agent that helps its user preserve its privacy. The privacy of users is preserved by a hybrid negotiation architecture where privacy domain and rules are represented semantically but the decision making is done by the agents using utility functions. The paper develops various negotiation strategies including one that exploits reciprocity. The key idea is that each agent keeps track of whether a certain other user has been helpful before in preserving privacy using a credit system. When agents help others in preserving their privacy, their credit increases so that later they can ask others to help them. Hence, helping others to preserve privacy serves as an incentive. Using these strategies, agents can negotiate on the content and agree on how it will be shared before the post goes online.

2 Negotiation Architecture

Our proposed negotiation architecture is based on semantic representation of negotiation concepts and privacy rules, but enables each agent to use its own utility functions to evaluate negotiation offers. We use PriNego [3] as the basis for the semantic aspects of negotiation. PriNego proposes a negotiation framework for privacy where each agent represents a user in the social network. Each agent is aware of the privacy concerns of its user but also has information about the social network, such as the friends of the user. This information is captured in an ontology that is represented in Web Ontology Language (OWL).

Privacy Concerns: Each agent captures its user's privacy concerns as semantic rules (privacy rules) represented with a Semantic Web Rule Language (SWRL) [1]. A privacy rule describes a situation wherein an agent would reject a particular negotiation offer. Consider a user Alice who does not want her colleagues to see her pictures within a leisure context. If Alice herself was sharing a picture in a leisure context, she or her agent can enforce that the audience of a post is set so that colleagues are not included. However, if Bob, a friend of Alice is about to share such a content, then it is difficult to enforce Alice's privacy constraint on Bob's content.

Negotiation: In such a setting, Alice and Bob's agents can negotiate among each other to decide if the content should be shared and if so, under which constraints. Following the above example, if Bob asks Alice to share a post in leisure context, then Alice's agent would reject this regarding Alice's privacy rule. Our proposed system enables agents to provide a rejection reason as well (e.g., rejected because of context). If a negotiation offer does not violate any of the privacy rules of the user, then the agent accepts this offer. For example, if Bob would ask Alice to share a post in work context, then Alice's agent would accept this offer. A user might have various privacy constraints but these might not be equally important. To capture the fact that a rule is more important than a second rule, we associate a weight with each rule.

Decision Making: When an agent creates an offer, the evaluations done to decide whether to accept an offer as well as to create a new counter-offer constitute the *negotiation strategy* of an agent. Here, we require each agent to have a utility function, which is based on the privacy concerns of the user. The utility value of a post request (negotiation offer) considers the threshold value set by the user agent as well. An agent makes a decision about a post request regarding its utility function.

The agent that initiates the negotiation (i.e., initiator) will

[1] Bogazici University, Turkey, email: dilara.kekulluoglu@boun.edu.tr
[2] Bogazici University, Turkey, email: nadin.kokciyan@boun.edu.tr
[3] Bogazici University, Turkey, email: pinar.yolum@boun.edu.tr

have a different utility function than an agent that negotiates for her privacy (i.e., negotiator) since both agents have different responsibilities during negotiation.

- A *negotiator agent* is responsible for evaluating a post request and making a decision about this post request based on its utility. In case where it wants to reject it, it may also provide rejection reason(s) depending on the strategy that it follows.
- An *initiator agent* is responsible for initializing the negotiation with other agents (i.e., agents that are relevant to a post request). Then, it collects responses from other agents. If all agents agree on sharing the post request, then it shares the post. Otherwise, it will try to update the post request according to its utility and rejection reasons of others. As a result of this, it can choose to share the post, continue or terminate the ongoing negotiation.

3 Strategies for Privacy Negotiation

We have developed three negotiation strategies that agents can use.

Good-Enough-Privacy (GEP): In this strategy, the initiator agent sends a post request to relevant agents. At each iteration, each agent provides a rejection reason if it rejects the post request. For this, each agent evaluates a post request by computing a utility. If this utility is above the agent's utility threshold, then the agent accepts the post request as it is. Otherwise, the agent finds its most important rule then it rejects the post request and provides the corresponding rejection reason.

Maximal-Privacy (MP): GEP strategy sends only one rule per iteration. However, the initiator agent may be willing to revise the post request by considering multiple rejection reasons. Hence, the negotiation could terminate in fewer iterations. For example, the initiator agent might want the negotiation to be over in two rounds, and an agent relevant to the post request might have three rules that are violated. The initiator agent may be actually ready to prevent all these violations. If the negotiator agent uses GEP strategy, then at most two rejection reasons can be considered. In MP, an agent will send all rejections reasons to the initiator agent. If the initiator agent rejects the post request, then the negotiator agent will start narrowing the set of rejection reasons by removing rejection reasons that are less important than others.

For GEP and MP, the outcome of the negotiation is only determined by considering the current situation and ignoring the previous interactions.

Reciprocal Strategy (RP): The outcome of a negotiation is beneficial for all the negotiating agents; however one party is usually better than the others. This difference might be insignificant for many negotiations. The difference may get disadvantageous for the others if one party is favored most of the times. To prevent this, we propose a new strategy based on reciprocity called *Reciprocal Strategy* (RP). In this strategy, agents negotiate regarding the previous behaviors and negotiations. If one party is favored more in previous negotiations, then this strategy tries to favor the other party. To keep track of the previous negotiations, we use a point-based system where both parties have the same amount of points in the initial state (e.g., each 5pts). For every negotiation, agents exchange points depending on who is the initiator and how much benefit they get from that negotiation.

At every negotiation iteration, the initiator agent sends the post request together with a point offer to the negotiator agent. In the previous strategies, the negotiator agent was calculating a utility per post request, and if this utility was below its utility threshold, it would send a rejection reason. In this strategy, agents also consider point offers of each other while computing their utilities. Hence, they try to compensate the utility shortage by the points that they get from others. If the computed utility is below the threshold, the negotiator agent asks the initiator agent for sufficient points to accept the post request. Otherwise, the negotiator agent accepts the post request as it is.

Since the negotiations in this strategy change depending on the previous interactions, the effects of RP should be captured observing continuous posting. We have tested our system with consecutive postings. In order to understand how posting habits of the people affect the outcome of the negotiation, we have tried different cases. To see how these posting habits affect the resulting utilities, we have considered various cases for two users: (i) one user shares a post, (ii) both users share posts regularly.

4 Future Directions

An important first step is to evaluate these strategies in comparison to each other. This comparison should take into account various factors such as overall points, time constraints, number of posts to be shared and so on.

As a second step, it is worthwhile to incorporate trust relations into the utility functions such that agents are more willing to cooperate with those that they trust. This would reflect real life relations more closely. Another important point is to enable the negotiation framework to be updated such that privacy rule weights can be learned over time.

ACKNOWLEDGEMENTS

This work has been supported by TUBITAK.

REFERENCES

[1] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al., 'SWRL: A semantic web rule language combining OWL and RuleML', *World Wide Web Consortium Member submission*, **21**, 79, (2004).

[2] N. R. Jennings, A. R. Lomuscio P. Faratin, S. Parsons, C. Sierra, and M. Wooldridge, 'Automated negotiation: Prospects, methods and challenges', *International Journal of Group Decision and Negotiation*, **10**(2), 199–215, (2001).

[3] Yavuz Mester, Nadin Kökciyan, and Pınar Yolum, 'Negotiating privacy constraints in online social networks', in *Advances in Social Computing and Multiagent Systems*, eds., Fernando Koch, Christian Guttmann, and Didac Busquets, volume 541 of *Communications in Computer and Information Science*, 112–129, Springer International Publishing, (2015).

[4] Jose M. Such and Michael Rovatsos, 'Privacy policy negotiation in social media', *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, **11**(1), 4:1–4:29, (2016).