Towards Secure Content Sharing in Social Networks

Clara Bertolissi
Aix-Marseille Univ., CNRS
Marseille, France
clara.bertolissi@lis-lab.fr

Alba Martinez Anton
Aix-Marseille Univ., CNRS
Marseille, France
alba.martinez-anton@lis-lab.fr

Romain Testud
Aix-Marseille Univ., CNRS
Marseille, France
romain.testud@lis-lab.fr

Nicola Zannone
Eindhoven Univ. of Technology
Eindhoven, The Netherlands
n.zannone@tue.nl

***Abstract*—We propose a solution for securing access to data shared on a community-centered collaborative platform, such as a Facebook style social network. Our solution leverages provenance information and social relationships between users to define a fine-grained access control model capturing users privacy preferences. We present a prototype implementation of our model and its validation on a case study.**

Keywords-Security; Privacy; Access control.

I. Introduction

In recent years, we have seen an increasing interest in community-centered collaborative systems, where users interact and provide access to a variety of information and personal data with different degrees of sensitivity [4]. In this context, balancing data sharing and security is a difficult problem. Traditional access control policies are insufficient for dealing with jointly-owned and jointly-managed content in such collaborative environments. In this work, we address the problem of secure sharing of information in social networks and discuss to what extent the access control mechanism provided by social networks allow users to control the permissions over the content they share.

Social networks, such as Facebook, offer their users an environment and functionalities to share contents with other users. These functionalities, however, can introduce privacy concerns for users, for instance when a user is tagged [2]. In this situation, the tag target has permissions to remove the tag from the content, but cannot modify the visibility of the tag. Methods for providing a fine-grained access control at the tag level are missing. The idea that inspired our approach is that the access control model should allow the tag target to specify its privacy preferences for tags and these preferences should be taken into account when displaying the photo to a requesting user.

Our framework considers a shared content as a compound object, possibly containing other objects as subcomponents. For instance, a photo may contain a list of comments and tags associated to it. Access requests are then evaluated considering the privacy settings not only of the requested object, but those of all its components. Moreover, we consider that the privacy settings, not only of the object owner, but also of other users involved in the creation of an object should be taken into account. Our solution uses a provenance data model inspired from the Open Provenance Model [3]. Using provenance information in the evaluation allows us to identify dependencies between objects (e.g., a comment associated to a photo in a post) and retrieve all users that are related to an object, either because they have triggered the process for creating the object or because they are involved in it with some specific role (e.g., host or tag target). We briefly describe our model in Section II, present a proof-of-concept prototype in Section III and apply it on a social case scenario in Section IV. We conclude in Section V.

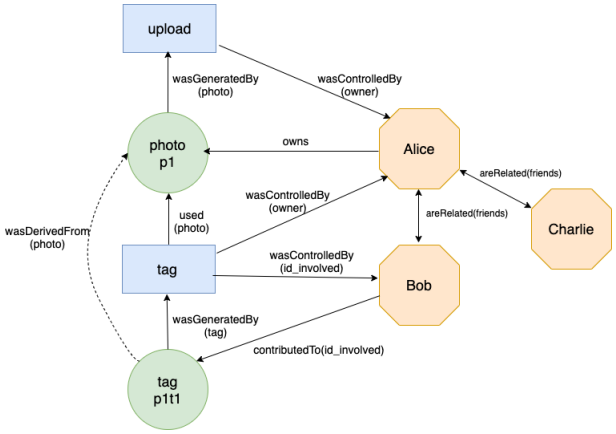

Fig. 1. Sample of the implemented evaluation graph

II. Extension of the Provenance model with user interaction dependencies

In this work, we leverage the Open Provenance (OP) Model [3] to capture causality dependencies between provenance entities within the social network, namely artefacts, agents and processes. Five main dependencies between two entities are defined: used (process on artifact), wasGeneratedBy (artifact on process), wasControlledBy (process on agent), wasDerivedFrom (artifact on artifact), and wasTriggeredBy (process on process). Altogether entities and dependencies form the nodes and the edges, respectively, of a directed acyclic graph. We apply the OP model to the setting of social networks, where artifacts are used to capture data objects (e.g., posts or comments); processes are used to capture the functional actions such as upload, comment, tag, etc.; agents correspond to the members of the social network. An example of provenance graph is presented in Fig. 1.

To reason about access constraints in a social network, we have extended the OP model by introducing new direct dependencies, such as owns (agent on artefact), and contributedTo (agent on process). Note that the OP model provides a notion of role that may be used to further specify dependencies. For example, an agent may have a dependency contributedTo(*id_involved*) with a tag artefact, meaning that the agent identifier has been used as tag target. We also introduce a NotAvaliable dependency which adds a cyclic dependency in

 ISBN: 978-1-61208-996-6

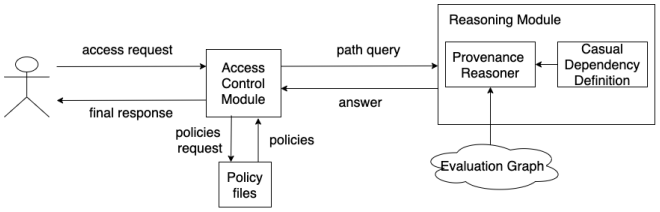

Fig. 2. Prototype Architecture

the provenance graph when an artefact has been deleted and is no longer visible.

In addition to provenance information, we also consider interpersonal social relationships between users, denoted by a dependency isRelated (and its symmetric closure areRelated) and characterized by a role such as *friend* or *family* specifying the nature of the relationship. Unlike provenance information, which represents the evolving of provenance data in the system, the social information shows a snapshot of the current relationships between users at a particular moment in time. Due to space limits, we depict the provenance and interpersonal dependencies as part of the same graph in Fig. 1.

III. Prototype: Architecture and Implementation

We describe the main components of our prototype, which are depicted in Fig. 2.

a) Access Control Module: The access control module intercepts access requests and computes the final access decision that is returned to the user. It uses auxiliary functions to retrieve user policies, to evaluate an access request and to pass path queries to the reasoning module.

b) User Policies: Each member of the social network can specify his privacy settings which are registered in a policy file. Following the Attribute-Based Access Control model, we define policies and access requests in terms of attribute name-value pairs, with the addition of path conditions (see [1]). User policies are defined in a simple language which provides an abstraction of the XACML policy specification language. For example, Alice can define her policy for objects of type *photo* as follows:

$$(\texttt{resource\_type} = photo \wedge \texttt{resource\_owner} = Alice$$
$$(\texttt{action} = comment \vee \texttt{action} = view)) \wedge$$
$$(\texttt{areRelated}(Alice, \texttt{user\_id}, friend)), \texttt{Permit})$$

meaning that a user can see and comment a photo owned by Alice if and only if he/she is a friend of Alice.

c) Reasoning Module: The Reasoner is implemented as a python program using the pyDatalog library. This allows us to benefit from the efficient reasoning capabilities provided by Datalog solvers for path condition resolution. In particular, we implemented rules both for constructing the evaluation graph from a given access log and for resolving path queries based on the obtained graph and the casual dependency definition.

d) Evaluation graph: The evaluation graph is built from the provenance information retrieved from the logs of the system together with the information about user relationships, which is updated at the moment of the access control request.

IV. Demonstration

Consider the social network represented by the OP model in Fig. 1 and assume an access request made by *Bob* to add a comment on Alice's photo $p1$. First, the request is received by the access control module. The object *owner* and *host*'s access policies are retrieved from the policy repository (they may be the same, say Alice policy, presented in the previous section). To evaluate the policy, the engine retrieves the regular path query from the policy $\texttt{areRelated}(Alice, \texttt{B}ob, friend))$ and pass it to the reasoning module to resolve it. If the answer is positive, the access control module returns a positive response to Bob: $[p1, permit]$. When necessary (e.g., host and owner are different users), a rule combining algorithm (we implemented, e.g., grant and deny override) is applied to combine multiple response into a final decision.

Consider now a request made by *Charlie* to view Alice's photo $p1$. The request is received by the access control module and treated as before. If the permission for viewing the photo is granted, the privacy settings associated to the comments and the tags attached to the photo are considered. In this case, the access control module interacts with the reasoning engine for retrieving all the users that contributed to the comments and tags of $p1$ (i.e., Bob for the tag $p1t1$ in Fig. 1) and gather the corresponding user policies. The request is thus decomposed in a list of access requests, each returning an access response for the associated object. The access control module gathers all the responses and compose them in the form of a list. Supposing Bob let only his friends view the tags where he is targeted, we obtain for our example $[[p1, permit], [p0c1, deny]]$, that is Charlie is granted access to view the photo but not the associated tag.

V. Conclusion

We have presented an approach that uses provenance data extended with social information for enabling a fine-grained access control mechanism implementing users privacy preferences. A proof-of-concept prototype is provided to demonstrate the feasibility of our approach. The integration of our work into an XACML architecture is left for future work. Path conditions can be specified in XACML policies using the element <PolicyIdReference> or they can be they can be encoded as user-defined functions.

Acknowledgment

The project leading to this publication has received funding from Excellence Initiative of Aix-Marseille - A*MIDEX (Archimedes Institute AMX-19-IET-009).

References

[1] C. Bertolissi, J. den Hartog, and N. Zannone, "Using provenance for secure data fusion in cooperative systems", In Proc. of SACMAT 2019, pp. 185-194. ACM, 2019.

[2] S. Damen and N. Zannone, "Privacy implications of privacy settings and tagging in Facebook", In Secure Data Management 2013, pp. 121-138.

[3] L. Moreau, et al., "The Open Provenance Model core specification," in Future Generation Computer Systems, 27(6), pp. 743–756, 2011.

[4] F. Paci, A. C. Squicciarini, N. Zannone, "Survey on Access Control for Community-Centered Collaborative Systems", ACM Comput. Surv. 51(1), pp. 6:1-6:38, 2018.

 ISBN: 978-1-61208-996-6