ARCHETYPAL MUTUAL CONTACT MECHANISM FOR ONLINE SOCIAL NETWORKS

Praneeth Kumar M*1, SKN Rehmatunnisa1

1M.Tech, Dept. of CSE, DJR Institute of Engineering & Technology, Gudavalli, Vijayawada (A.P), India.

ABSTRACT

Online social networks (OSNs) have marvelous growth in recent years and become genuine portals for hundreds of millions of Internet users. These OSNs offer attractive means for digital social interactions and information sharing, but also raise a number of security and confidentiality issues. While OSNs allow users to limit contact to shared data, they currently do not provide any mechanism to enforce confidentiality concerns over data associated with multiple users. To this end, we propose a mechanism to enable the protection of shared data associated with multiple users in OSNs. We formulate a contact control archetypal to capture the spirit of mutual approval necessities, along with a mutual mechanism requirement scheme and a mechanism enforcement mechanism. Also, we present a logical representation of our contact control archetypal that allows us to leverage the features of existing logic solvers to perform various analysis tasks on our archetypal. We also discuss a proof of concept archetypal of our approach as part of an application in Facebook and provide usability study and system appraisal of our method.

Keywords: Social network, mutual contact control, security archetypal, mechanism specification and management.

1. INTRODUCTION

ONLINE social networks (OSNs) such as Facebook, Google, and Twitter are essentially designed to enable people to share personal and public information and make social connections with friends, coworkers, colleagues, family, and even with strangers. In recent years, we have seen extraordinary growth in the application of OSNs. For example, Facebook, one of typical social network sites, claims that it has more than 860 million active users and over 30 billion pieces of content (web links, news stories, blog posts, notes, photo albums, and so on.) shared each month [1, 3]. To protect user data, contact control has become a central feature of OSNs [2, 4]. A typical OSN provides each user with a simulated space containing profile information, a list of the user’s friends and webpages, such as wall in Facebook, where users and friends can post content and leave messages. A user profile usually includes information with respect to the user’s birthday, gender, interests, education, and work history, and contact information. In addition, users can not only upload content into their own or others’ spaces but also tag other users who appear in the content. Each tag is an explicit reference that links to a user’s space. For the protection of user data, current OSNs indirectly require users to be system and mechanism managers for regulating their data, where users can contain data sharing to a specific set of trusted users. OSNs often use user relationship and group membership to distinguish between trusted and untrusted users. For example, in Facebook, users can allow friends, friends of friends (FOF), groups, or public to contact their data, depending on their personal authorization and privacy requirements.

Though OSNs currently provide simple contact control mechanisms allowing users to govern contact to info contained in their own spaces, users, unfortunately, have no control over data residing outside their spaces. For instance, if a user posts a comment in a friend’s space, she/he cannot state which users can view the comment. In another case, when a user uploads a photo and tags friends who appear in the photo, the tagged friends cannot restrict who can see this photo, even though the labeled friends may have different privacy concerns about the photo. To address such a critical issue, initial protection mechanisms have been offered by existing OSNs. For instance, Facebook allows tagged users to remove the tags linked to their profiles or report violations asking Facebook managers to remove the contents that they do not want to share with the public. However, these
simple protection mechanisms suffer from several limitations. On one hand, removing a tag from a photo can only prevent other members from seeing a user’s profile by means of the association link, but the user’s image is still contained in the photo. Since original contact control policies cannot be changed, the user’s image continues to be revealed to all authorized users. On the other hand, reporting to OSNs only allows us to either keep or delete the content. Such a binary decision from OSN managers is either too loose or too restrictive, relying on the OSN’s management and needful several people to report their request on the same content. Hence, it is vital to develop a real and flexible contact control mechanism for OSNs, obliging the special approval requirements coming from multiple associated users for managing the shared data collaboratively.

Pattern for profile and relationship sharing

We pursue a methodical solution to ease cooperative management of shared data in OSNs. We begin by investigating how the lack of mutual contact control (MAC) for data sharing in OSNs can undermine the defense of user data. Some typical data sharing patterns with respect to mutual authorization in OSNs are also identified. Based on these sharing patterns, an MAC archetypal is framed to capture the core types of mutual authorization requirements that have not been put up so far by existing contact control systems and archetypes for OSNs (e.g., [7, 8, 16, 9, 11]). Our archetypal also contains a mutual mechanism specification scheme. In the meantime, as conflicts are inevitable in mutual approval enforcement, a voting mechanism is further providing to deal with authorization and privacy conflicts in our archetypal.

Another convincing feature of our solution is the support of analysis on the MAC archetypal and systems. The correctness of implementation of a contact control archetypal is based on the premise that the contact control archetypal is valid. Furthermore, while the use of an MAC mechanism can greatly enhance the suppleness for adapting data sharing in OSNs, it may possibly reduce the certainty of system authorization costs due to the reason that authorization and privacy conflicts need to be resolved stylishly. Measuring the implications of contact control mechanisms traditionally relies on the security analysis technique, which has been applied in several domains (operating systems, trust management [12], and role-based contact control [6, 10]). In our methodology, we additionally introduce a method to represent and reason about our archetypal in a logic program. In addition, we provide an archetypal implementation of our authorization mechanism in the context of Facebook. Our trial results demonstrate the feasibility and usability of our method.

2. MAC FOR OSNs: NECESSITIES AND PATTERNS

We proceed with a comprehensive requirement analysis of MAC in OSNs. For now, we discuss some typical sharing patterns occurring in OSNs where multiple users may have different authorization requirements to a single resource. We exactly analyze three scenarios: profile sharing, relationship sharing, and content sharing—to understand the risks posted by the lack of collaborative control in OSNs. We leverage Facebook as the running example in our discussion because it is currently the most popular and representative social network provider. In the meantime, we reiterate that our discussion could be easily extended to other existing social network platforms, such as Google+ [4].

Profile sharing: An attractive feature of some OSNs is to support social applications written by third-party developers to make additional functionalities built on the top of users’ profile for OSNs [1]. To offer meaningful and attractive services, these social applications consume user profile properties, such as name, birthday, activities, interests, and so on. To make matters more complex, social applications on present OSN platforms...
can also consume the profile aspects of a user’s friends. In this case, users can select specific pieces of profile properties they are willing to share with the applications when their friends use the applications. At the same time, the users who are using the applications may also want to regulate what info of their friends is available to the applications since it is likely for the applications to conclude their private profile properties through their friends’ profile properties [13].

**Relationship sharing:** Another feature of OSNs is that users can share their relationships with other members. Relationships are inherently bidirectional and convey possibly sensitive info that associated users may not want to reveal. Most OSNs provide machineries that users can control the display of their friend lists. A user, still, can only control one direction of a relationship. Let us reflect, for example, a scenario where a user Amar specifies a mechanism to hide her friend list from the public. But, Bob, one of Amar’s friends, agrees a weaker mechanism that permits his friend list visible to anyone. In this case, if OSNs can solely enforce one party’s mechanism, the relationship between Amar and Bob can still be learned through Bob’s friend list. A relationship sharing form where a user called owner, who has an association with another user called shareholder, shares the relationship with a contactor. In this situation, authorization requirements from both the owner and the shareholder should be considered. Otherwise, the shareholder’s privacy concern may be violated.

**Content sharing:** OSNs offer built-in mechanisms enabling users to connect and share contents with other members. OSN users can post statuses and notes, upload photos and videos in their own spaces, tag others to their contents, and share the contents with their friends. On the other hand, users can also post contents in their friends’ spaces. The shared contents may be connected with multiple users. Consider an example where a photograph comprises three users, Amar, Bob, and Cindia. If Amar uploads it to his own space and tags both Bob and Cindia in the photo, we call Amar the owner of the photo, and Bob and Carol stakeholders of the photo. All of them may agree contact control policies to control over that can see this photo.

3. **MAC ARCHETYPAL FOR OSNs**

We solemnize an MAC archetypal for OSNs, as well as a mechanism scheme and a mechanism assessment mechanism for the specification and enforcement of MAC rules in OSNs.

3.1 **MAC Archetypal**

An OSN can be denoted by a relationship network, a set of user groups, and a gathering of user data. The relationship network of an OSN is a directed labeled graph, where each node denotes a user and each verge signifies a relationship between two users. The label associated with each verge indicates the type of the relationship. Verge direction denotes that the initial node of a verge launches the relationship and the terminal node of the verge accepts the relationship. The number and type of supported relationships rely on the specific OSNs and its purposes. Also, OSNs include an important feature that allows users to be systematized in groups [14, 15] or called circles in Google+ [5], where each group has an exclusive name. This feature enables users of an OSN to easily find other users with whom they might share specific interests (e.g., same hobbies), demographic groups (e.g., studying at the same schools), political alignment, and so on. Users can join in groups without any endorsement from other group members. Also, OSNs provide each member a web space where users can store and manage their personal data with profile information, friend list and content.
3.2 MAC Strategy Specification

To enable a collaborative authorization management of data sharing in OSNs, it is essential for MAC policies to be in place to regulate contact over shared data, representing approval requirements from multiple associated users. Our mechanism specification scheme is built upon the proposed MAC archetypal.

**Contactor specification:** Contactors are a set of users who are approved to contact the shared data. Contactors can be denoted with a set of user names, a set of relationship names (RNs) or a set of group names (GNs) in OSNs.

**Data specification:** In OSNs, user data are composed of three types of information: user profile, user relationship, and user content. To enable effective privacy conflict determination for MAC, we present sensitivity levels (SL) for data description, which are assigned by the controllers to the shared data items. A user’s judgment of the SL of the data is not binary (private/public), but multi-dimensional with varying degrees of sensitivity.

**Contact control mechanism:** To summarize the above-mentioned mechanism features, we present the definition of an MAC mechanism as follows:

Definition of MAC mechanism. A MAC mechanism is a 5-tuple \( P = \langle \text{controller, c type, contactor, data, effect} \rangle \).

The MAC policies

1. “Amar approves his friends to view his status known by status01 with a medium SL, where Amar is the owner of the status.”

2. “Bob approves users who are his colleagues or in hiking group to view a photo, winter.jpg, that he is tagged in with a high SL, where Bob is a stakeholder of the photo.”

3. “Cindia disallows Dave and Edward to watch a video, play.avi, that she uploads to someone else’s spaces with a highest SL, where Cindia is the provider of the video.”

3.3 Mutual Mechanism Evaluation

Two steps are done to evaluate a contact request over MAC policies. The first step checks the contact request against the mechanism stated by each controller and yields a decision for the controller. The contactor element in a mechanism decides whether the mechanism is applicable to a request. If the user who sends the request goes to the user set derived from the contactor of a mechanism, the mechanism is appropriate and the evaluation process yields a response with the decision (either permit or deny) indicated by the effected element in the mechanism. Otherwise, the response yields deny decision if the mechanism is not appropriate to the request. In the second step, decisions from all controllers responding to the contact request are amassed to make a final decision for the contact request. Since data controllers may generate different decisions (permit and deny) for a contact request, conflicts may happen. To make an unmistakable decision for each contact request, it is vital to adopt a methodical conflict resolve mechanism to resolve those conflicts during mutual mechanism evaluation.
A simple solution for deciding mutual privacy conflicts is to only allow the common users of contactor sets defined by the multiple controllers to contact the data item. Regrettably, this strategy is too limiting in many cases and may not harvest desirable results for resolving mutual privacy conflicts. Reflect an example that four users, Alice, Bob, Carol, and Dave, are the controllers of a photo, and each of them permits her/his friends to see the photo. Suppose that Alice, Bob, and Carol are close friends and have many common friends, but Dave has no common friends with them and also has a cute weak privacy concern on the photo. In this case, adopting the simple solution for conflict resolution may turn out that no one can contact this photo. Though, it is sensible to give the view permission to the common friends of Alice, Bob, and Carol.

3.3.1 A Voting System for Decision Making of Mutual Control

Mainstream voting is a popular instrument for decision making. Stimulated by such a decision-making mechanism, we suggest a voting scheme to achieve a real mutual conflict resolution for OSNs. A distinguished feature of the voting mechanism for conflict resolution is that the choice from each controller is able to have an effect on the final decision. Our voting scheme covers two voting mechanisms: decision voting and sensitivity voting.

Assume that all controllers are similarly important; an aggregated decision value (DVag) (with a range of 0.00 to 1.00) from many controllers with the owner (DVow), the contributor (DVcb), and stakeholders (DVst) is computed with following equations:

\[ DV_{ag} = \left( DV_{ow} + DV_{cb} + \sum_{i \in SS} DV_{si} \right) \times \frac{1}{m} , \]

\[ DV_{ag} = \left( \frac{1}{\omega_{ow} + \omega_{cb} + \sum_{i=1}^{n} \omega_{si}} \times DV_{ow} \times DV_{cb} + \sum_{i=1}^{n} (\omega_{si} \times DV_{si}) \right) \]

**Sensitivity voting:** Each controller assigns an SL to the public data item to reflect her/his privacy worry. A sensitivity score (Sc) (in the range from 0.00 to 1.00) for the data item can be considered based on following equation:

\[ Sc = \left( SL_{ow} + SL_{cb} + \sum_{i \in SS} SL_{si} \right) \times \frac{1}{m} , \]

3.3.2 Verge Based Conflict Resolution

An elementary idea of our method for verge based conflict resolution is that the Sc can be used as a verge for decision making. Instinctively, if the Sc is higher, the final decision has a high chance to deny contact, taking into account the privacy protection of high sensitive data.

3.3.3 Strategy Based Conflict Resolution with Privacy Recommendation

Above threshold-based conflict resolution delivers rather fair machinery for making the final decision when we treat all controllers equally significant. Though, in practice, different controllers may have different primacies in making impact on the final decision. In specific, the owner of data item may be desirable to possess the highest priority in the control of shared data item. Thus, we additionally provide a strategy based conflict determination mechanism to fulfill precise authorization requirements from the owners of common data.

3.3.4 Conflict Determination for Diffusion Control

A user can share others’ data with her/his friends in OSNs. In this case, the user is a disseminator of the content, and the content will be posted in the disseminator’s space and visible to her/his friends or the public. Since a disseminator may adopt a weaker control over the disseminated content but the content may be much more
From the perspective of original controllers of the content, the confidentiality worries from the original controllers of the content should be always fulfilled, averting inadvertent revelation of sensitive contents.

4. RATIONAL PICTURE AND ANALYSIS OF MAC

We adopt answer set programming (ASP), a recent form of declarative programming, to properly represent our prototypical, which fundamentally provide a formal foundation of our archetypal in terms of ASP-based representation. Then, we prove how the correctness analysis and authorization analysis [7] of MAC can be approved based on such a rational picture.

4.1 Representing MAC in ASP

We present a conversion module that turns mutual authorization specification into an ASP program. This interprets a formal semantics of mutual authorization specification in terms of the answer set semantics.

4.1.1 Rational Definition of Mutual Controllers and Transitive Relationships

The basic components and relations in our MAC archetypal can be directly demarcated with matching predicates in ASP. We have defined UDct as a set of user-to-data relations with controller type ct 2 CT. Then, the logical definition of multiple controllers is as follows:

- The owner of a data item can be represented as:  
  OW(controller, data) ← UDct(controller, data) ∧  
  U(controller) ∧ D(data).

- The contributor of a data item can be represented as:  
  CB(controller, data) ← UDct(controller, data) ∧  
  U(controller) ∧ D(data).

- The stakeholder of a data item can be represented as:  
  ST(controller, data) ← UDct(controller, data) ∧  
  U(controller) ∧ D(data).

- The disseminator of a data item can be represented as:  
  DS(controller, data) ← UDct(controller, data) ∧  
  U(controller) ∧ D(data).

5. EXECUTION AND ASSESSMENT

5.1 Archetype Execution

We executed a proof-of-concept Facebook application for the cooperative management of shared data, called MController (http://apps.facebook.com/MController). Our archetypal application enables multiple associated users to specify their approval policies and privacy preferences to control a common data item. It is value noting that our current execution was limited to handle photo sharing in OSNs. Adversely, our approach can be generalized to deal with other types of data sharing, such as videos and comments, in OSNs as long as the stakeholder of shared data is recognized with real methods like tagging or searching.

MControl application structure
Decision making in MControl

A core element of MController is the decision-making module, which procedures contact requests and returns responses (either permit or deny) for the requests. The figure showed system architecture of the decision-making module in MController. To evaluate a contact request, the policies of each controller of the targeted content are imposed first to generate a decision for the controller. Before, the decisions of all controllers are amassed to yield a final choice as the response of the request.

User interface of MControl

5.2 System Usability and Performance Assessment

5.2.1 Participants and Process

MController is a useful proof-of-concept application of mutual privacy management. To measure the practicality and usability of our mechanism, we conducted a survey study (n = 35) to explore the issues surrounding users’ desires for confidentiality and discover how we might recover those applied in MController.
Precisely, we were interested in users’ viewpoints on the current Facebook privacy system and their desires for more control over photos they do not own. We recruited participants through university mailing lists and through Facebook itself using Facebook’s built-in sharing API. Users were given the chance to share our application and play with their friends. While this is not a random sampling, employing using the natural distribution features of Facebook debatably gives a precise profile of the ecosystem.

5.2.2 User Study of MController

For assessment determinations, questions (http://goo.gl/eDkaV) were divided into three areas: likeability, simplicity, and control. Likeability is a degree of a user’s satisfaction with a system (e.g., “I like the idea of being able to control photos in which I am tagged”). Simplicity is a measure how instinctive and useful the system is (e.g., “Setting my privacy settings for a photo in MController is Complicated (1) to Simple (5)” with a 5-point scale). Control is a measure of the user’s apparent control of their individual data (e.g., “If Facebook implemented controls like MController’s to control photo privacy; my photos would be better protected”). Questions were either True/False or restrained on a 5-point Likert scale, and all replies were scaled from 0 to 1 for numerical analysis. In the extent, a higher number indicates a positive insight or opinion of the system while a lower number designates a negative one. To examine the average user awareness of the system, we used a 95 percent confidence interval for the users’ answers. This assumes the population to be mostly normal.

5.2.3 Performance Assessment

To assess the performance of the policy assessment mechanism in MController, we changed the number of the controllers of a shared photo from 1 to 20, and allocated each controller with the average number of friends, 130, which is appealed by Facebook statistics [3]. Also, we considered two cases for our evaluation. In the first case, each controller permits “friends” to contact the shared photo. In the second case, controllers specify “FOF” as the contactors instead of “friends.” In our experiments, we performed 1,000 autonomous tests and measured the performance of each trial. Since the system performance depends on other processes running at the time of dimension, we had initial inconsistencies in our performance.

6. DISCUSSION

In our MAC system, a group of users could conspire with one another so as to influence the final contact control decision. Study an attack scenario, where a set of malevolent users may want to make a shared photo available to a wider audience. Supposing they can contact the photo, and then they all tag them or fake their individualities to the photo. In addition, they conspire with each other to assign a very low SL for the photo and stipulate policies to grant a wider audience to contact the photo. With a large number of conspiring users, the photo may be disclosed to those users who are not predictable to gain the contact. To stop such an attack scenario from happening, three conditions need to be satisfied: 1) there is no bogus identity in OSNs; 2) all tagged users are real users seemed in the photo; and 3) all controllers of the photo are authentic to specify their privacy partialities.

7. CONNECTED WORK

Contact control for OSNs is still a comparatively new research area. Numerous contact control archetypal for OSNs have been introduced (e.g., [7, 8, 9, 11, 16]). Early contact control solutions for OSNs introduced trust-based contact control enthused by the developments of trust and status computation in OSNs. The D-FOAF system [11] is mainly a friend of a friend ontology-based disseminated identity management system for OSNs,
where relations are related with a trust level, which shows the level of friendship between the users par taking in a given relationship.

8. CONCLUSION

In this paper, we have projected a new solution for cooperative management of shared data in OSNs. An MAC archetypal was articulated, along with a mutual policy condition scheme and corresponding policy assessment mechanism. In addition, we have presented a method for expressing and cognitive about our proposed archetypal. A proof-of-concept application of our solution called MController has been discussed as well, trailed by the usability study and system evaluation of our technique.

REFERENCES