Towards Secure Cooperation in Online Social Networks

Youna Jung
ACIS, Department of Electrical and Computer Engineering
University of Florida, Gainesville
FL 32611, USA
younajung@ufl.edu

Minsoo Kim
5065 NW 45th Road, #102
Gainesville, FL 32606, USA
minsoo.ai@gmail.com

James B.D Joshi
LERSAIS, School of Information Science
University of Pittsburgh, Pittsburgh,
PA 15260, USA
jjoshi@pitt.edu

***Abstract*—The rapid growth of online social networks (OSNs) has brought a revolutionary change in the way geographically dispersed people interact and cooperate with each other towards achieving some common goals. Recently, some new ways of ad-hoc cooperation have been demonstrated during the hurricane Irene and the earthquake in Japan. In such emergency situations, OSNs have already taken a significant role as alternative social media that support altruistic information sharing and cooperation among people. However, existing cooperation approaches have not been well-organized and are highly vulnerable to security threats such as a disclosure of users' identities and the leakage of other private data due to the lack of secure cooperation mechanisms. To support secure and effective cooperation in OSNs, in this paper, we propose the Social CRiBAC (Community-centric Role interaction Based Access Control) model, which extends the existing CRiBAC model [1] for use in OSNs to support cooperation among users. To verify the feasibility of the proposed model, we have implemented a prototype application on Facebook and demonstrate its applicability with two working examples.**

Keywords- Online Social Network, Community, Cooperation, Anonymous Member, Temporal Sharing, Property-based Access Control

I. Introduction

The rapid advances in networking technologies have significantly enhanced the level of connectivity and interactions among people around the world. It has resulted in the explosive growth of online social networks (OSNs) such as Facebook and Twitter. People advertise themselves, make new friends, and maintain their relationships through OSNs. According to Facebook's statistics [2], 1 in every 10 people on earth uses Facebook, with over 750 million users. Over 375 million of them (over 50%) log in every day and spend over 700 billion minutes per month there.

The rapid growth of OSNs has significantly influenced human lifestyles, especially in the patterns of communication and cooperation. Since OSNs can provide a huge pool of manpower and support quick diffusion of information, they can be a basis for immediate and effective cooperation among people. These OSNs allow a large number of users who are globally dispersed to connect to each other, thus, providing an unprecedented opportunity to enhance the level of social cooperation towards achieving some common goals. In fact, several real cases support this claim. In some recent emergency situations related to natural disasters, OSNs have provided huge support for global level social activities. When Hurricane Irene occurred and tsunami hit Japan, the OSN users shared critical information about Irene's path and evacuation plans, and also warned residents about possible damages [3]. In the immediate aftermath of the earthquake and tsunami in Japan, most of the infrastructure was destroyed and mobile phones were largely silenced. In such a disaster situation, OSNs such as Facebook and Twitter took the place of traditional media and communication infrastructure to report the updated news and became the best link between worried family members and their friends and loved ones [4]. To help the victims more concretely and in an organized way, aid organizations also rushed to use OSNs to be aware of real situations, let people know what victims need, and recruit volunteers. As an example, the Red Cross raised relief funds via Twitter to assist tsunami victims around the Pacific Rim. In addition, it piloted a new program to engage digital volunteers on Twitter to help the victims with rescue and recovery efforts during Hurricane Irene [5]. It posted the requirements and recruited volunteers through these digital infrastructures. Thus, OSNs have played an important role in emergency situations not only as an alternative media that collects and spreads valuable information, but also as an effective way to gather people and encourage them to cooperate with each other to achieve a common goal.

There are also several other application scenarios which show us the usefulness of cooperation through OSNs. Recently, many people are using Twitter to find their lost pets by broadcasting information about them [6][7]. Another example is from the healthcare domain where the use of an OSN resulted in an immediate cooperation among people [8]: a medical doctor who works for Emory Healthcare received a tweet from a person named Matthew about an emergency situation involving his grandmother. A medical team communicated with Matthew via Twitter to instruct him to give first aid while emergency transportation was being arranged. The doctor said, "Without the quickness of social media, the helicopter may have never been dispatched".

As seen in the aforementioned examples, OSNs have an enormous potential for helping people by supporting dynamic and real-time cooperation among users. Although some real cases show significant promise for success, it is not easy to guarantee effective and secure cooperation in existing OSNs

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States
Copyright © 2012 ICST
DOI 10.4108/icst.collaboratecom.2012.250707

due to the lack of a suitable cooperation model and an appropriate security model integrated with it for ONSs. To date, cooperation among OSN users has been achieved in an ad hoc manner only. Without a cooperation model and OSNs' systematic support, it is difficult to expect successful cooperation in OSNs. Furthermore, the absence of suitable security model for cooperation in OSNs can bring OSNs to a serious security crisis. As an example, allowing accesses to private information and resources during cooperation may raise serious privacy and security/safety problems. To facilitate immediate and secure cooperation in OSNs, in this paper, we propose an access control model which meets the security requirements for OSNs and support effective and secure cooperation among users.

The paper is organized as follows. In Section 2, we present our motivation using two examples and identify the security requirements for secure cooperation in OSNs. In Section 3, we propose a community-centric access control model for OSNs called Social CRiBAC and present an OSN employing the proposed model. To illustrate the feasibility and practical use of the proposed model, in Section 4, we demonstrate a Facebook application and show how it meets the security requirements identified in Section 2. In Section 5, we discuss related work, and finally, we present the conclusions and future work in Section 6.

II. MOTIVATION

From ancient times, people have been cooperating with each other to cope with difficulties. Recently, OSNs have promoted cooperation among people as we can see in the real cases mentioned earlier. In this section, we overview existing cooperation approaches in OSNs using two example scenarios and discuss its limitations and derive the security requirements.

A. *Motivating Examples*

For better understanding of our motivation, we describe two motivating examples: *Disaster Relief* and *Finding a Lost Child*. Both show emergent situations that require immediate, well-organized, and secure cooperation among OSN users.

1) Disaster Relief: Many people need an immediate help in disaster situations such as hurricanes, earthquakes, and a terrorist attack. To rescue victims quickly and effectively, it is required to provide well-organized and practical help immediately. Let's assume that a man is injured in a disaster situation. He posts on an OSN using his mobile phone to ask for help (D1). Many people who see the post spread it to let more people know his urgent situation and some volunteers go out to rescue him (D2). Shortly afterward, other victims including a seriously injured woman also ask for help in many different places simultaneously (D3). To rescue as many victims as possible, the most suitable rescue team should be organized for every victim. To do so, it is necessary to form a cooperative group of volunteers who are close to a victim and are capable of giving necessary aid including medical aid. If volunteers flock to a few victims who have posted earlier, other victims may not be rescued. Even though a victim has many helpers, the lack of vital aid may lead to a failure in rescue work.

2) Finding a Lost Child : As we mentioned earlier, there several real-world examples where OSN users have cooperated to find lost pets [6][7]. With regards to finding a lost child, however, we must show proper discretion in posting information of a lost child in public. Let's assume a mother has lost her daughter at a children's festival held in a crowded place in a city. She posts her daughter's photo and identity on an OSN to receive help while she is waiting for policemen to arrive (F1). Some of OSN users who are near the place and see the post may try to find the lost child (F2). If someone finds the lost girl, he or she can take care of the girl and let her mother know via the OSN (F3). Broadcasting the lost girl's information and photos may help to find her but it may cause very serious security and safety problems also if the information is seen by bad people.

B. *Limitation of Existing Cooperation in OSNs*

As can be seen in the examples above, the OSNs are able to promote cooperation among users by spreading information quickly. However, cooperation achieved in such OSNs seems to be ad hoc in nature and support provided by OSNs for such cooperation is limited. Furthermore, its vulnerability to security threats is a barrier to further progress. In this section, we identify the limitations of existing cooperation in OSNs as follows.

1) Limited help: When a user posts an urgent message on an OSN, the message is visible to only some users who are connected to him, for examples, his friends, friends of friends, and followers. If he opens the post to the public, more people who randomly visit his page are able to see the posting, even if they do not have any social relationships with him. However, there is no way to let others who are not related to him and also never visit his page know about his emergency situation. In other words, he is unable to receive help from them, even if one of them is the most suitable one who can give him some vital help immediately. We therefore need to find a way to make the best use of manpower in OSNs more effectively.

2) Naïve cooperation: As we pointed out in the *Disaster Relief* example, the lack of cooperation mechanism may lead to failure in cooperation. In the D3 situation, many victims shall be confronted by death if rescue teams are not well-organized based on helpers' locations and abilities. In addition, in the F2 situation, the mother may be able to find her daughter more efficiently if the search areas for each volunteer can be assigned based on the volunteers' locations. Therefore, to ensure the success of a cooperation in an urgent situation it is necessary to have a suitable cooperation model which supports well-organized cooperation among OSN users.

3) Vulnerability of private information and data: In urgent situations, a user seeking prompt help may rush to share private information and data with unknown people without configuring proper privacy settings. However, such indiscriminate sharing may raise serious security and privacy problems. For examples, in D2 situation, it may be inappropriate to share the victim's medical record with other volunteers, except for a particular helper who can give medical aid, due to the possibility of misuse. Similarly, it is risky as well as useless to share the lost girl's information with bystanders. From the perspective of the volunteers, some of them may not want to reveal their private information such as location and participation. Their concern about privacy may make them unwilling to help people. To remove such a concern, a cooperative group should be formed with only necessary and trustworthy users and sharing of their private information should be properly controlled.

C. *Security Requirements for cooperation in OSNs*

As discussed above, the existing OSNs do not have support for enabling a more structured, goal-driven cooperation among users. Moreover, the ad hoc cooperation that occurs can have severe security and privacy vulnerabilities that create barriers to their practical use. Especially, their vulnerabilities create concrete barriers to their practical use and effectiveness. In the existing literature, many researchers have specified security objectives for OSNs to protect users' information [9][10], and derived security requirements such as anonymity, personal information/data protection, consistency between OSN users and real-world users, high availability of services and data in OSNs, and so on [10][11]. In this paper, we focus on the access control requirements. Researchers have identified the unique requirements for access control in OSNs as follows.

1) *Consideration of social relationship* [11, 12, 13, 14, 15] – The access control models/systems must be able to control accesses based on the social relationships among OSN users.
2) *Fine-grained control* [11, 14, 15] – The access control models should be able to independently provide a fine granularity of control on personal information or data.
3) *Individualized policies* [16] – The access control models should allow individualized policies for each user rather than one system-wide policy.
4) *Sticky policy with the users' data* [11][16] – The access control models should allow specifying a sticky policy for each data item.
5) *Interoperability* [11] – The access control policies should be usable across multiple OSNs.
6) *Users and users' policies as the target of control* [14, 15, 16] – The access control models should be capable of controlling the activities on other users and the accesses to their policies.
7) *OSN's behavior control* [16] – The access control models must be able to control the OSNs' behavior as well as OSN users' behavior.

So far, there, however, exists no approach that considers cooperation among users. In this paper, we identify the requirements for secure cooperation in OSNs as follows.

1) *Organization of a community which consists of eligible users only* – To prevent unnecessary sharing of members' private information and resources and follow the principle of least privilege, a community must be organized with only eligible users and the members' resources should be shared with only particular members who need the resources to cooperate with other members.
2) *Anonymous cooperation* – The OSNs must guarantee anonymous participation in a community of users, where necessary, and also facilitate sharing of resources anonymously for privacy protection.
3) *Time-based control* – All of shared resources and granted permissions for cooperation must be valid only during cooperation. When a community is disorganized or a member is no longer available, all permissions for its cooperation must be revoked immediately.

III. ACCESS CONTROL MODEL FOR EFFECTIVE AND SECURE COOPERATION IN OSNS

To meet the requirements for secure cooperation in OSNs including general security requirements, in this section, we propose an access control model, called Social CRiBAC, which allows well-organized and secure cooperation in OSNs and provides two case studies based on the motivating examples introduced in Section 2.

A. *Preliminaries*

Before we propose Social CRiBAC, we first introduce the CRiBAC (Community-centric Role interaction based Access Control Model) [1], as the proposed Social CRiBAC model extends it into the domain of OSNs. CRiBAC aims to support secure cooperation within a community, as well as interactions between highly heterogeneous and decentralized agents. In CRiBAC, a community is dynamically organized to achieve its goal. It adopts the community computing model (CCM) [17] to create a community with only eligible agents and also helps facilitate efficient cooperation by employing a situation-aware cooperation model. In addition, it incorporates interaction permissions and community-related entities: such as society, community, community role, and so on, in order to control accesses to agents' own objects, tasks, and agents themselves during interaction and cooperation. Fig. 1 illustrates the basics of the CRiBAC model.

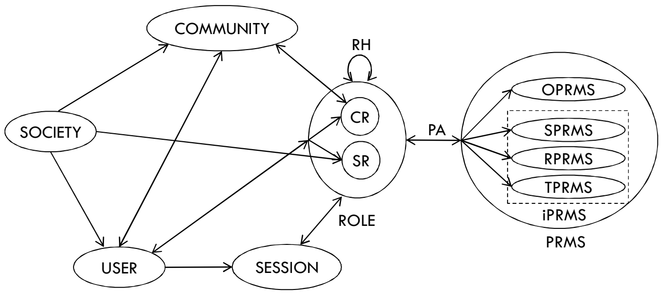


Figure 1. CRiBAC Model

In CRiBAC, an agent has its own resources, contexts, and tasks. An agent's resources are a set of objects which belong to the agent and access to them is controlled by that agent. An agent also has its own contexts that capture specific information such as status and identification, and its own tasks to show what kinds of work it can do. An agent can participate in one or more communities by assuming various community roles (*CR*). A community *c* consists of necessary CRs (CR_c), participating agents (A_c), and its context ($CONT_c$) which represents information related to *c* such as its type, creation time, and information about members. A community type specifies its goal, necessary roles (*CR*), and a set of policies to assign agents to a CR (*CRA*). According to CRA policies, a community invites suitable agents for each CR and then selects the most appropriate agents based on their contexts and tasks. After receiving an invitation, every agent must decide whether it can participate or not. This decision is made by an invited agent itself based on its ability and preference. If an agent accepts the invitation and is finally selected, it assumes the suggested CR and starts cooperation with other members. The CR is revoked from a member agent when the community is terminated or it leaves the community.

To control interaction and cooperation among agents as well as accesses to objects, CRiBAC defines two types of permissions: traditional *object-oriented permissions* (*OPRMS*) and *interaction permissions*. The interaction permissions include the resource-oriented permissions (*SPRMS*), the role-oriented permissions (*RPRMS*), and the task-oriented permissions (*TPRMS*). A *sprms* is a permission that allows access to agents' resources. An *rprms* is a permission that allows a subject agent to carry out its task on another target agent. A *tprms* is a permission that allows a subject agent to command a target agent to perform a task of the target agent. Interaction or cooperation among agents can be authorized only if they have corresponding interaction permissions. Some permissions are parameterized based on roles. The parameterized permissions should be should be enabled based on the parameter values associated with the real agents who are assigned to the corresponding role, after the role assignment.

B. *Social CRiBAC*

Existing CRiBAC model does not fit OSNs due to the lack of consideration for unique characteristics of OSNs, although it has a definite advantage of ensuring secure interaction and cooperation. To guarantee secure cooperation in OSNs, in this paper, we propose a property-based access control model, called Social CRiBAC, which meets the security requirements identified in Section 2.

In Social CRiBAC, a user has four types of properties; 1) contexts (*u.CNT*) which represent a user *u*'s status such as age, sex, job, reputation, and social relationships; 2) tasks (*u.TSK*) which *u* can carry out in an OSN such as data uploading and posting messages, 3) resources (*u.RSC*) which *u* creates in his private space in an OSN such as photos and postings, and 4) policies (*u.POL*). The user policies have two types: the access control policy (*u.AC*) and the filtering policy (*u.FP*). The access control policy is to decide authorized users to his own properties while the filtering policy is to filter unwanted contents out from all of authorized contents using his preferences.

A community is a mission-oriented cooperative group of users who are eligible and willing to cooperate with other members. As an occasion demands, a community is dynamically created and terminated when its goal is achieved. A community has five properties: 1) contexts (*c.CNT*) such as community goal, 2) tasks (*c.TSK*) such as community creation and termination, 3) resources (*c.RSC*) shared with members, 4) a cooperation (*c.COP*) which describes cooperative process among members using the situation-based cooperation model [17] that describes members' tasks according to a community's situation, and 5) policies (*c.POL*). A community has two types of policies: the access control policy (*c.AP*) and the recruiting policy (*c.RCP*). The access control policy is used to authorize accesses to the properties of other members and the recruiting policy specifies the eligibility rule for each community role.

A society represents an OSN supporting secure cooperation, and it also has four properties: 1) contexts (CNT_s) such as the number of users, 2) Tasks (TSK_s) that represent an OSN's services to promote social activities among users such as displaying of friends who are online and notifying friends' recent news, 3) resources (RSC_s) that are available to all the users, and 4) policies (POL_s) that are enforced on all users; for example, users who are under age are prohibited to access to the contents tagged '*adult only*'. A permission consists of an operation and one or more objects. In Social CRiBAC, various properties can be a target object. According to the types of the object, the permissions are categorized into five types: *context-oriented permission* (*CP*), *tasks-oriented permission* (*TP*), *resources-oriented permission* (*RP*), *policies-oriented permission* (*PP*), and *users-oriented permission* (*UP*). A user can have a permission through the permission assignment (*PA*). A community member who participates in a community can have the required permissions to cooperate with others by taking a community role (*CRA*) because all necessary permissions are assigned to the CR through the CPA assignment. The formal definition of Social CRiBAC is shown in Table I.

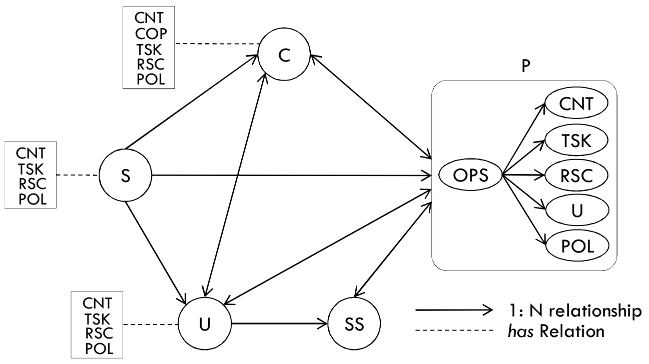


Figure 2. Social CRiBAC

Social CRiBAC meets the security requirements for access control in OSNs, which includes not only general access control but also cooperation control. We present how the proposed model satisfies the requirements described in Section 2 in details.

1) *Consideration of social relationship* – In Social CRiBAC, interpersonal relationships are represented in a user's contexts and also used for making decisions on access requests as one of the criteria.
2) *Fine-grained control* – Each property items including private information and resources can be separately specified and controlled at a fine-grained level.
3) *Individualized policies* – Each user has his own policy (*user policy*).
4) *Sticky policy with the users' data* – The access control policy of a user is used for specifying sticky policies on the user's own properties.

5) *Interoperability* – Social CRiBAC can be used in many different OSNs since it considers not only social graph but also diverse properties which users and communities have.
6) *Users and users' policies as the target of control* – To control the accesses to other users and their policies, Social CRiBAC uses the user-oriented permissions and the policy-oriented permissions.
7) *OSN's behavior control* – A user can control an OSN's service and control behavior by having own filtering policy and the task-oriented permissions for OSN's tasks.
8) *Organization of a community which consists of eligible users only* – Using the recruiting policy, a community can have only eligible users.
9) *Anonymous cooperation* – Social CRiBAC allows a user to participate in a community under an alias to protect members' privacy (*Anonymous participation*). In addition, members share their private information and resources by posting/uploading them on a community space which only authorized members can access. By doing this, members do not need to establish social relationships between them to share their resources so it is hard to identify the real owners of the community resources (*Anonymous resource*).
10) *Time-based restriction* – All permissions on a community's properties is valid only while the community is alive and an authorized user is available.

TABLE I. FORMAL DEFINITION OF SOCIAL CRiBAC

	Definition	**Description**
CNT	$CNT_s \cup CNT_C \cup CNT_U$	The set of all possible contexts which represent an OSN (CNT_s), communities (CNT_C), and users (CNT_U). e.g., in Facebook, CNT_U ={*Basic-Info, Friend, Family, Edu, Work, Philosophy, Arts, Sports, Activities, Contact, …*}
TSK	$TSK_s \cup TSK_C \cup TSK_U$	The set of all tasks that an OSN (TSK_s), its communities (TSK_C), and users (TSK_U) can carry out within an OSN.
RSC	$RSC_s \cup RSC_C \cup RSC_U$	The set of all resources that a society (RSC_s), its communities (RSC_C), and users (RSC_U) own in an OSN. A resource can have a set of attributes and we use dot notation to represent attributes associated with a resource (*rsc.att*). Formally, $RSC_C = \{c_i.RSC \mid 1 \le i \ge m \text{ where } c_i \in C\}$, $RSC_U = \{u_j.RSC \mid 1 \le j \ge n \text{ where } u_j \in U\}$. e.g). In Facebook, RSC_U ={*Wall-posting, Photo, Video, Note*}, $u_j.RSC.Photo=\{Photo_k \mid 1 \le k \ge p \text{ where } Photo.att= (Date, TaggedPerson)\}$, $u_j.Photo_k.TaggedPerson=$"*Jane*"
POL	$POL_s \cup POL_C \cup POL_U$	The set of all policies of an OSN (POL_s), communities (POL_C), and users (POL_U).
u	$<u.CNT, u.TSK, u.RSC, u.POL>$	A user u in U has own contexts ($u.CNT \subset CNT_U$), tasks ($u.TSK \subset TSK_U$), resources ($u.RSC \subset RSC_U$), and policies ($u.POL \subset POL_U$). Formally, $u.POL= u.AP \cup u.FP$ is a set of access control policies for a user. $u.AP$ is a set of access control policies on a user u's properties. Formally, $u.AP$: $cond_u \rightarrow PA_u$ where $cond_u$ is a property-based user predicate. $u.FP$ is a set of filtering policies of u, formally, $u.FP$: $cond_c \rightarrow hide(contents)$ where $contents =\{CNT \mid RSC \mid TSK_s\}$and $(u,contents) \in PA$.
CR		The set of all community roles in an OSN. Each user participating in a community has to take one or more CRs, but all CRs should be revoked from members when the community is terminated. CR_c is the set of CRs involving in a community c.
U_c	$U_c \subseteq U$	The set of users who participate in a community c, called *community member.*
U_{cr}	$U_{cr} \subseteq U$	The set of users taking a community role cr.
OBJ	$OBJ_{CNT} \cup OBJ_{TSK} \cup OBJ_{RSC} \cup OBJ_{POL} \cup OBJ_U$	The set of all target objects of OPS, which should be protected from unauthorized access.
OBJ_{CNT}	$OBJ_{CNT} \subseteq CNT$	The set of all contexts that can be accessed by users.
OBJ_{TSK}	$OBJ_{TSK} \subseteq TSK$	The set of all tasks that can be a target object of a task-oriented permission.
OBJ_{RSC}	$OBJ_{RSC} \subseteq RSC$	The set of all resources that can be accessed by users.
OBJ_{POL}	$OBJ_{POL} \subseteq POL$	The set of all policies that can be accessed by users.
OBJ_U	$OBJ_U \subseteq U$	The set of users that can be an object of a user's operation.
OPS		The set of all applicable operations on OBJ.
CP	$OPS \times OBJ_{CNT}$	The set of all context-oriented permissions. CP_c is a set of all CPs whose target contexts are the contexts of a community c. Formally, $CP_c= OPS \times c.CNT$ where $c.CNT \subset OBJ_{CNT}$. CP_u is a set of all CPs whose target contexts are the contexts of a user u. Formally, $CP_u= OPS \times u.CNT$ where $u.CNT \subset OBJ_{CNT}$.
TP	$OPS \times OBJ_{TSK}$	The set of all task-oriented permissions. $TP_c= OPS \times c.TSK$ where $c.TSK \subset OBJ_{TSK}$ and $TP_u= OPS \times u.TSK$ where $u.TSK \subset OBJ_{TSK}$.
RP	$OPS \times OBJ_{RSC}$	The set of all resource-oriented permissions. $RP_c= OPS \times c.RSC$ where $c.RSC \subset OBJ_{RSC}$ and $RP_u= OPS \times u.RSC$ where $u.RSC \subset OBJ_{RSC}$.
PP	$OPS \times OBJ_{POL}$	The set of all policy-oriented permissions, called *User Admin Permission*, which allow users to admin other users' policies. $PP_c= OPS \times c.POL$ where $c.POL \subset OBJ_{POL}$ and $PP_u= OPS \times u.POL$ where $u.POL \subset OBJ_{POL}$.
UP	$OPS \times OBJ_U$	The set of all user-oriented permissions which allows a user to carry out its operation on a target user. $UP_c= OPS \times U_c$ where $U_c \subset U$ and $UP_u= OPS \times u$ where $u \subset U$.
P	$CP \cup TP \cup RP \cup PP \cup UP$	A set of permissions in an online society.
PA	$\{CPA \cup TPA \cup RPA \cup PPA \cup UPA\} \subseteq \mathrm{U} \times \mathrm{P}$	A many-to-many user to permission assignment relationship, where $CPA \subseteq U \times CP$, $TPA \subseteq U \times TP$, $RPA \subseteq U \times RP$, $PPA \subseteq U \times PP$, *and* $UA \subseteq U \times UP$. $PA_c= CP_c \cup TP_c \cup RP_c \cup PP_c \cup UP_c$ is a set of PAs whose target objects are a community c's properties. $PA_u= CP_u \cup TP_u \cup RP_u \cup PP_u \cup UP_u$ is a set of PAs whose target objects are a user u's properties.
CRA		A many-to-many user to community role assignment relationship, where $CRA \subseteq U \times CR$. $CRA_c= U \times CR_c$.
SS		The set of all sessions created for users in an OSN.
S	$<CNT_s, TSK_s, RSC_s, POL_s, C, U, PA>$	A CS-OSN s represents an OSN and it has a set of society contexts ($CNT_s \subset CNT$), a set of society tasks ($TSK_s \subset TSK$) a set of society resources ($RSC_s \subset RSC$), and a set of society policies ($POL_s \subset POL$), a set of communities (C), a set of users (U), and a set of permission assignment relationships (PA).
c	$<c.CNT, c.TSK, c.RSC, c.COP, c.POL, c.CRA>$	A community has a set of community contexts ($c.CNT \subset CNT$), a set of community tasks ($c.TSK \subset TSK$), a set of community resources ($c.RSC \subset RSC$), a cooperative process ($c.COP$), a set of user assignments to CR_c ($c.CRA=CR_c \times U_c$.), and a set of community policies ($c.POL=\{c.AP \cup c.RCP\} \subset POL$). $c.AP$ is the set of access control policies for a community c's properties. Formally, $c.AP: cond_u \rightarrow PA_c$. $c.RCP$ is the set of recruiting policies for c. Formally, $c.RCP: cond_u \rightarrow CRA_c$ where $CRA_c \in CRA$. $c.COP$ is specified as a partially ordered set of community members' tasks ($U_c.TSK \subset TSK_U$). Formally, $c.COP = \{S_i: (U_c.TSK, \le) \mid 1 \le i \le n\}$.

C. Secure Cooperation-supporting OSN (SeCON)

A SeCON is an OSN which supports prompt and secure cooperation among users by employing Social CRiBAC. In a SeCON, every user who is willing to give and/or take help through communities must let the system know his availability and preferences by setting his corresponding properties such as location, job, online status, and access control policies in advance of actual cooperation. Fig. 3 presents the overview of a SeCON.

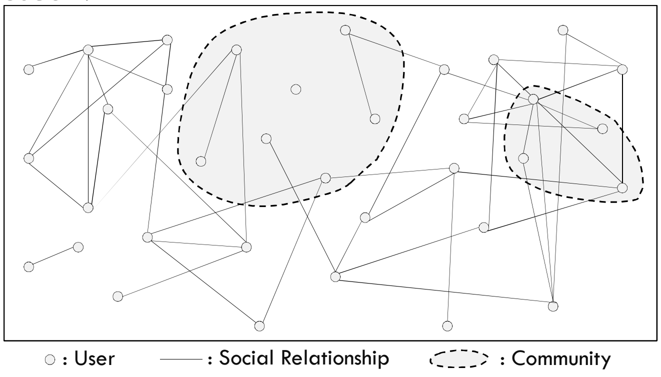


Figure 3. Overview of SeCON

If a user needs help, he has to inform a SeCON of his situation and desire. To help him, the system first creates a community space and then recruits the most eligible and suitable users based on their properties according to a community's recruiting policies (*c.RCP*). After having all necessary members, a SeCON grants the required permissions to each of the members according to a community's access control policies (*c.AP*) to ensure successful cooperation. To achieve a common goal, all the members undertake given tasks and cooperate with each other according to a cooperative process (*c.COP*). After achieving the goal, the community is dissolved and all permissions which relate to the cooperation are revoked. By employing Social CRiBAC, a SeCON is able to guarantee the security and privacy protection requirements not only in ordinary cases but also during cooperation by supporting organization of a trustworthy community with only eligible users while supporting anonymous cooperation and time-based sharing.

D. Case Studies

In this section, we describe two communities by using Social CRiBAC based on the motivating examples.

1) Finding a Lost Child:

*c*1 = [(*goal*, *Finding a lost child*),{*createCom*, *terminateCom*}, {*childIdentity*, *childPhoto*, *helperlocation*, *searchArea*, *search Result*}, {S_1, S_2, S_3, S_4}, {AP_1, AP_2}, {RCP_1, RCP_2}, {(*parent*, *Alice's mom*), (*police*, *P*1), (*helper*,{*H*1, *H*2, *H*3, *H*4})}], where

- S_1: *parent*.*createRSC*(*c*1.*RSC*.*childIdentity* & *c*1.*RSC*.*child Photo*) & *helper*.*createRSC* (*location*, *c*1.*RSC*.*helperlocation*)
- S_2: *police*.*createRSC*(*c*1.*RSC*.*searchArea*)
- S_3:*helper*.*createRSC*(*c*1.*RSC*.*searchResult*)
- S_4: *searchResult*= "Found"$\rightarrow$ *Termination*
- AP_1: *has_cnt*(u_s.*CRs* $\ni$*c*1.*parent*)$\rightarrow$ *RPA*(u_s,(*write*, *c*1.*childIdentity*&*childPhoto*&*searchResult*)) & *TPA* (u_s, (*request*, *c*1.*terminateCom*))
- AP_2: *has_cnt*(u_s.*CRs* $\ni$ *c*1.*police*) $\rightarrow$ *RPA*(u_s,(*read*, *c*1.*childIdentity*&*childPhoto*&*helperlocation*))& *RPA*(u_s, (*execute*, *c*1.*searchArea*)) & *TPA* (u_s, (*request*, *c*1.*terminateCom*))
- AP_3: *has_cnt*(u_s.*CRs* $\ni$ *c*1.*helper*) $\rightarrow$ *RPA*(u_s, (*read*, *c*1.*childIdentity*&*childPhoto*& *searchArea*)) & *RPA* (u_s, (*write*, *c*1.*helperlocation*&*searchResult*))
- RCP_1: *has_cnt*(u_s.*affiliation*="*Police*") $\rightarrow$ CRA(u_s,*c*1.*Police*)
- RCP_2: *has_cnt*(u_s.*location*= *the place where a mother lost a child*) & *has_cnt*(u_s.*reputation* $\geq$ *the required reputation*) $\rightarrow$ *CRA*(u_s, *c*1.*helper*)}

2) Disaster Relief:

*c*2=[(*goal*, *Rescue a patient who has cardiac disease*), {*createCom*, *terminateCom*}, {*patientMedicalHistoty*, *patientLocation*, *patientMedicalSituation*, *firstaidInstruction*, *rescueSituation*}, {S_1, S_2, S_3, S_4}, {AP_1, AP_2, AP_3,}, {RCP_1, RCP_2}, {(*parent*, *P*1), (*cardiologist*,*D*1), (*helper*,{*H*5,*H*6})}], where

- S_1: *patient*.*createRSC*(*c*2.*RSC*.*patientLocation*& *patient MedicalSituation*)
- S_2: *helper*.*createRSC*(*c*2.*RSC*.*patientMedicalSituation*)
- S_3: *cardiologist*.*createRSC*(*c*2.*RSC*.*firstaidInstruction*)
- S_4: *rescueSituation* ="*Rescued*" $\rightarrow$ *Termination*
- AP_1: *has_cnt*(u_s.*CRs* $\ni$ *c*2.*patient*) $\rightarrow$ *RPA*(u_s, (*write*,*c*2.*RSC*.*patientLocation*& *patientMedical Histoty*) & *TPA* (u_s, (*request*, *c*2.*terminateCom*))
- AP_2: *has_cnt*(u_s.*CRs* $\ni$ *c*2.*helper*) $\rightarrow$ *RPA*(u_s, (*read*, *c*2.*patientLocation*)) & *RPA*(u_s, (*write*, *c*2.*patientMedicalSituation*)) & *TPA* (u_s, (*request*, *c*2.*terminateCom*))
- AP_3: *has_cnt*(u_s.*CRs* $\ni$ *c*1.*cardiologist*) $\rightarrow$ *RPA*(u_s, (*read*, *c*2.*patientLocation*&*patientMedicalHistoty*& *patientMedicalSituation*) & *RPA* (u_s, (*read*&*write*, *c*2.*firstaidInstruction*))
- RCP_1: *has_cnt*(u_s.*affiliation*="*Cardiologist*") $\rightarrow$ *CRA*(u_s,*c*1. *cardiologist*),
- RCP_2: *has_cnt*(u_s.*location*= *the place closed to a patient*) & *has_cnt*(u_s.*reputation* $\geq$ *the required reputation*) $\rightarrow$ *CRA*(u_s, *c*2.*helper*)}

IV. IMPLEMENTATION

In this section, we demonstrate a SeCON application in Facebook, called SeCON app, and present how it supports efficient and secure cooperation among users by using the working example based on the 'Finding a lost child' scenario.

A. Facebook Application of SeCON

We have implemented a prototype of SeCON app which is a Facebook application that guarantees secure cooperation among the Facebook users. As a test bed for evaluating our work, we chose Facebook to take its advantages. As the most popular OSN, first, Facebook allows the SeCON app to have a huge pool of potential collaborators so a user is able to easily get necessary help by using the abundant human resources in Facebook. Second, Facebook supports rich interaction and cooperation among users through wall postings, messages, and group organization. By using such communication methods, we can promote cooperation among Facebook users. Third, Facebook provides many useful APIs to the public for developing applications on Facebook, hence it helps us to reduce the time and efforts to implement an application. In fact, the SeCON Facebook Application (in short, SeCON App) is developed as a web application and utilizes several Facebook APIs, as follows. For social plug-in, we use *Like Button*, *Comments*, *Wall Postings*, and *Activity Feeds*; for Authentication, we use *Login* and *Registration*. The main page of the SeCON App introduces the objectives of the application and available community services. This page is opened to the public but a user should register with the SeCON app to use community services. When registering, a user needs to allow the app to access the user's private information in Facebook such as profile, check-ins, online status, and location. Getting a user's context is essential for the SeCON App to select suitable members for a particular community.

The architecture of the SeCON App and the SeCON engine is presented in Fig.4. A Facebook user accesses the SeCON App to ask other users for help. After creating a community, members send requests to the SeCON engine to access the community's properties and other members' properties. The SeCON engine deals with accesses according to the defined Social CRiBAC policies. The request created by the user is sent to the Policy Enforcement Point (PEP). The PEP Module then delivers the request to the Policy Decision Point (PDP) with information about the member and his request. The PDP Module decides whether or not to grant the required permission. In order to do so, the PDP fetches the corresponding policies from Policy Repository that has Social CRiBAC policies created by the system administrators, and evaluates them by using the relevant properties retrieved from the Property Repository. After making a decision, the PDP conveys the result to the PEP, and the PEP performs authorization by granting the requested permissions. If necessary, the PEP in the SeCON engine can collaborate with the PEP of Facebook to control accesses to particular properties of Facebook users, such as wall postings, photos, and messages.

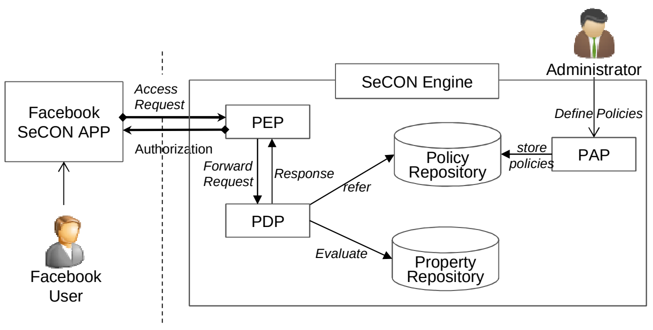


Figure 4. Architecture of SeCON App and SeCON engine

B. *Cooperation in the SeCON Application*

In this section, we minutely describe a cooperation process of the SeCON App in the perspective of users.

1) *Request for a community service:* A user who wants to receive a certain community service pushes the '*Community Creation Request*' button on the app and then selects the most appropriate community template and specifies requirements for cooperation. Note that we assume that this app possesses sufficient templates of communities, which have information about cooperation and policies in order to reduce the time to create a community.

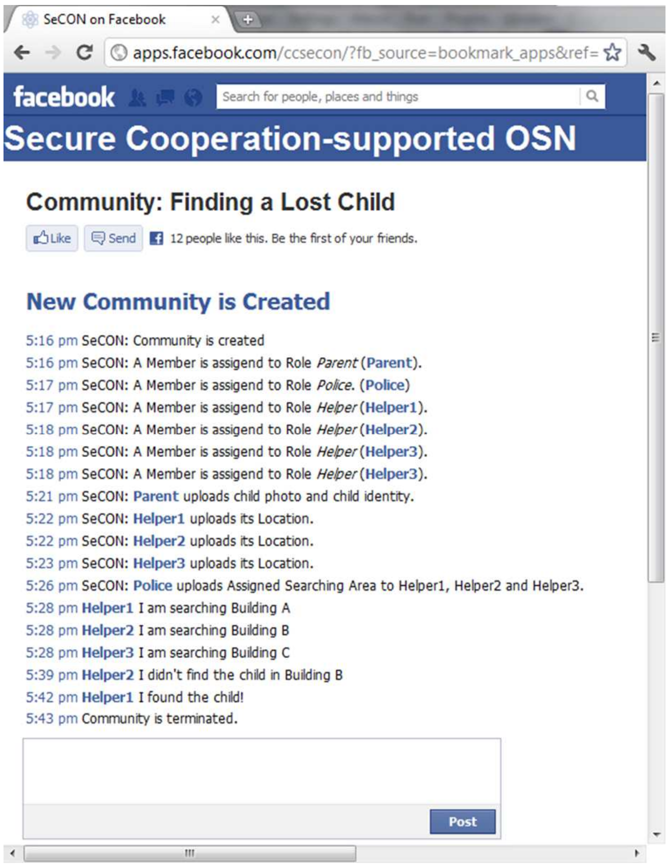
2) *Creation of an online community:* When the SeCON app receives a request for a community creation, then it creates a space for a community and invites the suitable candidates for each community role by using the context information of registered users, such as sex, age, work, location, and reputation. To invite a user, the SeCON app posts an invitation on the user's wall and then he can accept or reject the community role offered.

3) *Orchestration of cooperation among members:* After all members have been recruited, they start to cooperate with each other to achieve a common goal. A community manager created by the app facilitates cooperation by sending messages to the members the tasks that each member has to perform according to the community's situation. Each member performs the assigned tasks and posts the results on the community's wall. The community manager also controls accesses to the community's resources using the community's policies during cooperation. It distributes the community policies to members to let the members control the accesses to their own properties from other members. To illustrate the situation-aware cooperation process let's consider the '*Finding a lost child*' scenario in detail. In S_1, the SeCON app sends a message to the mother of a missing girl to let her upload her daughter's identity information and a photo as the community's resources with a proper role-oriented permission, *RP*(*write,c1.childIdentity & childPhoto*). At the same time, all the helpers who have *RP*(*write, c1.*u_s*.location*) can write their location in the community's *location*. In S_2, the SeCON app sends a message to a policeman to arrange the search areas for each helper using the members' location information and then lets all the helpers read the assigned search area by accessing the community resource. In S_3, each helper needs to post a search result. If someone finds the lost girl, the community's situation is changed into S_4 and a policeman can ask the SeCON app to terminate the community (as the goal has been met) with a permission, *TP* (*request, c1.terminateCom*).

4) *Community dissolution:* Once the community's goal is achieved, all community roles are revoked from members and the permissions related to community roles are consequently revoked. Finally, the community manager deletes the space for the community in the SeCON App.

For better understanding, we present example cooperation in SeCON app based on the '*Finding a lost child*' scenario. (1)

A mother who lost her daughter requests a cooperative help through the SeCON app by selecting a community template of '*Finding a lost one*'. (2) The SeCON app gets information of community roles from the template and then finds suitable candidates for each role based on the users' context information. All the members who are eligible and willing to participate in the community can be assigned to one or more community roles. (3) After a community is created, its members can access the community's resources such as a lost girl's identity and photo. For efficient searching, *P1*, a policeman who is in charge of this mission, assigns a search area to each helper. To do so, *P1* can access the location information of all helpers. Each member posts the search result on the community wall that is visible to members only. If a member finds the lost child, he reports it and the initiator (in this case, the mother of a lost child) informs the community manager. (4) Then, the community is dissolved and all permissions that were assigned to members for this cooperation are revoked. It means that the members are no longer able to access the community resources and other members' properties. Finally, the community space is deleted from the SeCON App. The working example of a '*Finding a lost child*' community in SeCON App is presented in Fig. 5.


SeCON on Facebook
apps.facebook.com/ccsecon/?fb_source=bookmark_apps&ref=
facebook
Search for people, places and things

Secure Cooperation-supported OSN

Community: Finding a Lost Child

Like Send 12 people like this. Be the first of your friends.

New Community is Created

```
5:16 pm SeCON: Community is created
5:16 pm SeCON: A Member is assigend to Role Parent (Parent).
5:17 pm SeCON: A Member is assigend to Role Police. (Police)
5:17 pm SeCON: A Member is assigend to Role Helper (Helper1).
5:18 pm SeCON: A Member is assigend to Role Helper (Helper2).
5:18 pm SeCON: A Member is assigend to Role Helper (Helper3).
5:18 pm SeCON: A Member is assigend to Role Helper (Helper3).
5:21 pm SeCON: Parent uploads child photo and child identity.
5:22 pm SeCON: Helper1 uploads its Location.
5:22 pm SeCON: Helper2 uploads its Location.
5:23 pm SeCON: Helper3 uploads its Location.
5:26 pm SeCON: Police uploads Assigned Searching Area to Helper1, Helper2 and Helper3.
5:28 pm Helper1 I am searching Building A
5:28 pm Helper2 I am searching Building B
5:28 pm Helper3 I am searching Building C
5:39 pm Helper2 I didn't find the child in Building B
5:42 pm Helper1 I found the child!
5:43 pm Community is terminated.
```

Figure 5. Example of a '*Finding a lost child*' community in the SeCON Application

V. RELATED WORK

To encourage many users to cooperate with each other in OSNs, we should be able to guarantee the security and privacy [18]. In fact, security is one of the most challenging issues in the social computing research. To protect a user's private information and online properties, an OSN needs an appropriate access control model. Many researchers have recently proposed access control models for the OSNs. Most of the proposed models have been focused on the social relationships among users and/or between users and users' own resources in OSNs [11, 12, 13, 14, 15].

Gates [11] has emphasized importance of interpersonal relationship to control accesses in OSNs and propose the relationship-based access control (ReBAC). Fong et al. have focused on the expressiveness of the ReBAC policies and proposed several policy languages. Those models employ the social relationships among users as the basis for authorization decisions and their policy languages are based on modal logic [12] and/or hybrid logic [13] to express complex policies.

Carminati et al. [14] have proposed a relationship-based access control model for OSNs. In this model, three types of security policies are proposed: 1) access control policy to control the accesses to resources including the users' resources, 2) filtering policy to specify how resources have to be filtered out when a user fetches his page, and 3) admin policy to decide who is authorized to specify policies. For the flexibility and interoperability of the access control model for OSNs, the proposed model uses semantic web technologies. The social network knowledge base (SNKB) is used to model users' profiles and actions, relationships between users and resources as well as relationships among users, and resources. The security policies are modeled using OWL and SWRL. For practical use of the model, authors present enforcement architecture and experimental results in [15].

For the fine-grained and systematic access control in OSNs, Park et al. [16] have proposed the activity-centric access control model for social computing (ACON). The ACON introduces a new term, *activity* which refers to the behaviors of users and a social network system. By using *activity*, ACON is able to control a system's automatic and administrative services as well as users' actions. An *activity* consists of action(s) and target(s), and users can be a target as well as resources. In addition, ACON allows individuals to have their own policies to specify privacy preferences and multiple sessions concurrently with different policies and attributes. This model is a relationship-independent access control model and it can capture more diverse aspects of OSNs using the attributes of users, administrators, a social network system, resources, and sessions. Many existing access control models mentioned earlier have tried to meet the security requirements of OSNs but there is no work that considers interactions and cooperation in OSNs. We present a comparison with existing models and Social CRiBAC in Table II.

TABLE II. COMPARISON WITH EXISTING MODELS

	Fong et al.	*Carminati et al.*	*ACON*	*Social CRiBAC*
Target objects	Resources	Resources Policies	Resources Users Attributes Policies	Resources Users Contexts Tasks Policies
Authorization	Relationship-dependent	Relationship-dependent	Relationship-independent	Relationship-independent
Interaction control	No	No	Limited[a]	Yes[b]
Cooperation control	No	No	No	Yes[c]

a. Able to control the role-oriented interactions.
b. Interaction Permissions: the role-oriented permission and task-oriented permission.
c. Cooperation model, anonymous cooperation and temporal privilege.

VI. CONCLUSION

The incredible growth of OSNs has promoted rich interactions and dynamic cooperation among users in emergency situations. However, support for cooperation in existing OSNs is still in its infancy due to lack of an effective cooperation mechanism and a security model. To overcome the limitations, we have identified the security requirements for efficient and secure cooperation in OSNs and have proposed an access control model, called Social CRiBAC, which meets the requirements. We summarize the major contributions of this paper as follows.

- We have identified the security requirements for efficient and secure cooperation among users in OSNs.
- We have proposed Social CRiBAC to control unauthorized accesses to the properties of users, communities, and an OSN such as context information, tasks, resources, and policies. The proposed model enables OSNs to guarantee efficient organization and secure cooperation of communities.
- We have proposed a Secure Cooperation-Supported OSN (SeCON) that employs Social CRiBAC and have presented two case studies.
- We have implemented a Facebook application, called the SeCON app, to show the feasibility of Social CRiBAC and SeCON based on an example case study, *Finding a lost child*. Through the implementation, we have demonstrated the potential of the proposed work.

To guarantee more reliable and dynamic cooperation in OSNs, some future research directions include the following:

- An administration model to facilitate administration of the social CRiBAC policies.
- Security analysis on the social CRiBAC model.

ACKNOWLEDGEMENT

The work of James Joshi has been supported by the by the US National Science Foundation award IIS-0545912.

REFERENCES

[1] Y. Jung and J. B. D. Joshi, “CRiBAC: Community-centric role interaction based access control model”, Computers & Security, vol. 31 No.4, 2012, pp. 497-523

[2] Facebook Statistics, http://www.facebook.com/press/ info.php?statistics

[3] Fox News, http://www.myfoxdfw.com/dpps/news/people-react-to-irene-on-facebook-and-twitter-dpgoh-20110829-fc_14765896

[4] ABC News, ‘Japan Earthquake and Tsunami: Social Media Spreads News, Raises Relief Funds’, http://abcnews.go.com/ Technology/japan-earthquake-tsunami-drive-social-media-dialogue/story?id=13117677

[5] Volunteer on Twitter to Help with Hurricane Irene and Other Disasters, http://hope140.org/blog/?p=209

[6] Lost dog found on Twitter, http://twitter.com/#!/ TheLDFBand

[7] Fidofinder on Twitter, http://twitter.com/#!/fidofinder

[8] Can Twitter Help Save Lives? A Health Care Social Media Case Study-Part I, http://advancingyourhealth.org/highlights /2011/04/27/can-twitter-help-save-lives-a-health-care-social-media-case-study-part-i/

[9] L. A. cutillo, R. Molva, and T. Strufe, “Safebook: A privacy-preserving online social network leveraging on real-life trust”, IEEE Communications, Vol. 47, No. 12, Dec 2009, pp. 94 - 101

[10] C. Zhang, J. Sun, X. Zhu, and Y. Fang, “Privacy and security for online social networks: challenges and opportunities”, IEEE Network, vol 24, No. 4, July-August 2010, pp 13 - 18

[11] Carrie E. Gates. Access Control requirements for Web 2.0 security and privacy. In IEEE Web 2.0 privacy and security wjorkshop (W2SP’07), Oakland, California, USA, may 2007

[12] Philip W.L. Fong and Ida Siahaan, 2011. Relationship-Based Access Control Policies and Their Policy Languages. In Proceeding of the 16th ACM symposium on Access control models and technologies (SACMAT’11). ACM New York. 51-60

[13] Glenn Bruns, Philip W. L. Fong, Ida Siahaan, Michael Huth. Relationship-based access control: its expression and enforcement through hybrid logic. In Proceeding of the 2nd ACM Conference on Data and Application Security and Privacy (CODASPY 2012), San Antonio, USA, February 7-9, 2012. 117-124

[14] Barbara Carminati, Elena Ferrari, and Andrea Perego. 2009. Enforcing access control in Web-based social networks. ACM Transactions on Informationand System Security (TISSEC). Vol. 13, No. 1, Article 6, November 2009, 1-38

[15] Barbara Carminati, Elena Ferrari, Raymond Heatherly, Murat Kantarcioglu, Bhavani Thuraisingham. Semantic web-based social network access control. Computers & Security, Vol. 30, No. 2-3. (08 March 2011), pp. 108-115

[16] J. Park, R. Sandhu, and Y. Cheng, “ACON: Activity-Centric Access Control for Social Computing”, the 6th International Conference on Availability, Reliability and Security (ARES), 22-26 Aug. 2011, pp. 242 - 247

[17] Y. Jung and M. Kim, “Situation-Aware Community Computing Model for Developing Dynamic Ubiquitous Computing Systems”, Journal of Universal Computer Science, Vol. 16, No.15, 2010, pp. 2139-2174.

[18] M. Parameswaran, “Social computing: An overview”, Communications of the Association for Information Systems, Vol.19, 2007, pp.762-780.