11-24-2014 12:00 AM

A Collaborative Access Control Model for Shared Items in Online Social Networks

Hanaa Al Shareef, *The University of Western Ontario*

Supervisor: Sylvia Osborn, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Recommended Citation

Al Shareef, Hanaa, "A Collaborative Access Control Model for Shared Items in Online Social Networks" (2014). *Electronic Thesis and Dissertation Repository*. 2535.
https://ir.lib.uwo.ca/etd/2535

A COLLABORATIVE ACCESS CONTROL MODEL FOR SHARED ITEMS IN ONLINE SOCIAL NETWORKS

(Monograph)

by

Hanaa Alshareef

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

The recent emergence of online social networks (OSNs) has changed the communication behaviors of thousand of millions of users. OSNs have become significant platforms for connecting users, sharing information, and a valuable source of private and sensitive data about individuals. While OSNs insert constantly new social features to increase the interaction between users, they, unfortunately, offer primitive access control mechanisms that place the burden of privacy policy configuration solely on the holder who has shared data in her/his profile regardless of other associated users, who may have different privacy preferences. Therefore, current OSN privacy mechanisms violate the privacy of all stakeholders by giving one user full authority over another's privacy settings, which is extremely ineffective. Based on such considerations, it is essential to develop an effective and flexible access control model for OSNs, accommodating the special administration requirements coming from multiple users having a variety of privacy policies over shared items.

In order to solve the identified problems, we begin by analyzing OSN scenarios where at least two users should be involved in the access control process. Afterward, we propose collaborative access control framework that enables multiple controllers of the shared item to collaboratively specify their privacy settings and to resolve the conflicts among co-controllers with different requirements and desires. We establish our conflict resolution strategy's rules to achieve the desired equilibrium between the privacy of online users and the utility of sharing data in OSNs. We present a policy specification scheme for collaborative access control and authorization administration. Based on these considerations, we devise algorithms to achieve a collaborative access control policy over who can access or disseminate the shared item and who cannot. In our dissertation, we also present the implementation details of a proof-of-concept prototype of our approach to demonstrate the effectiveness of such an approach. With our approach, sharing and interconnection among users in OSNs will be promoted in a more trustworthy environment.

Acknowledgments

I thank Allah first and foremost for his blessings, reconciling me and enabling my thesis to come to light.

I would like to start by thanking the people without whom I would not have been able to write my thesis. My parents, Zana and Amer Alshareef, I sincerely thank you for your unconditional love and countless support throughout my life and especially during my master. From each of you, I learned to always seek knowledge, break my limits and your supportive words and prayers helped me to overcome the challenges I encountered and gave me the strength to succeed. To you both I dedicate this thesis.

I would like to express my deepest gratitude to my supervisor, Dr. Sylvia Osborn, for her unwavering support and encouragement over the years. Her guidance, thoughtful advice, insightful comments, continuous understanding and kind patience helped me to improve myself, advance my knowledge and achieve my goals successfully.

My sisters and brothers: I thank you all for your kind wishes, supportive words and providing such a loving environment. You were a source of inspiration to me throughout the years. A big thanks goes to my friends, especially Wafaa Alshareef, Hussein, Amal and Tagreed who were closest to me, forgave much, put up with much and loved much.

Finally, I would like to extend my sincerest thanks to King Khalid University and the Saudi Arabian Cultural Bureau in Canada for generously supporting and encouraging me toward excellence.

Table of Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

Online Social Networks (OSNs), a very popular application on the Internet, have attracted almost one billion users, many of whom have incorporated these applications into their daily practices [Deep Nishar. April 18, 2014, Twitter Inc. June 2014, Socialbakers. 2014, Google Official Blog. April 11, 2012]. Nowadays, there are hundreds of OSN sites which facilitate and enable the users to interact and collaborate with each other in a virtual community. The rapid rise of a large variety of OSN sites, with the massive amount of available information, obviously raises new, serious concerns about the security and privacy of their users and requires insights into security and privacy issues. Researchers from different computer science disciplines have investigated some of the privacy and security problems which arise in OSNs, from different viewpoints (e.g., [Raad and Chbeir. 2013, Pesce, et al. 2012, Hu and Ahn. 2011, Gurses and Diaz. 2013, Hongyu Gao, et al. 2011, Mahmood. 2013]). As a shared platform, the lack of collaborative policy for access control and authorization management has become one of the most important and crucial issues in OSNs. Currently, OSNs have limited access control where the privacy settings of shared content is solely defined and regulated by the uploader of the shard content, regardless of other involved users. Hence, because of the limited and poor access control mechanisms for shared data in OSNs, the concerns of information disclosure are increased. We believe improving OSN access control models by devising a collaborative policy and management for it emerges as the first step toward tackling the existing security and privacy concerns related to online social networks.

The recent emergence of online social networks (OSNs) has transformed the World Wide Web from an information pool to a platform for communication and social interaction. OSN sites (e.g., Facebook, LinkedIn, Flickr, Twitter, etc.) provide an environment and massive types of services to clearly encourage users to socialize and interact with each other both on the Web and in the real world. These applications offer massive types of tools for their users (e.g., posting, tagging, uploading, commenting, re-sharing, etc.) to share information (e.g., photos, contacts, interests, activities, backgrounds, etc.). Also, in OSNs, users can build their profiles and begin social relationships with each other for a variety of purposes (e.g., business, entertainment, dating etc.). As a result, OSNs have become the most successful and most widely used services on the Web. According to a report by Socialbakers.com, Facebook has almost 1 billion users [Socialbakers. 2014]. In April 2014 LinkedIn reports on its website they have 300 million members from more than 200 countries and territories[Deep Nishar. April 18, 2014]. Additionally, Twitter has 230 million active users (as of June 2014), tweeting an average of 500 million tweets every day[Twitter Inc. June 2014]. Moreover, Google+ has 170 million active users (as of April 2012) [Google Official Blog. April 11, 2012]. Several statistics reveal that OSNs have become one of the highest used web applications in our lives all over the world [Goel, et al. 2012]. A Nielsen study shows that online users often are willing to spend their web time on social networks and blogs[Nielsen. 2012].

The recent popularity and adoption of OSN sites produce more and more information that is publicly available on the Web and easily accessible from anywhere. The availability of these vast amounts of personal information within OSNs obviously raises security and privacy concerns and issues. Actually, it is clear that sharing personal information, photos and other contents are the main purpose of OSNs. Therefore, as a shared platform, data in OSNs may be co-controlled by a number of users, just as books can be co-authored. Such co-control occurs by using different tools in OSNs such as posting, tagging or re-sharing. OSNs' users have an unclear idea about who can view their shared information and whose privacy policies govern the sharing of their information. Let us discuses the activity of photo sharing and tagging that is one of the most popular features of OSNs and has often become part of personal identity management [Besmer and Richter Lipford. 2010]. For example, assume Alice took pictures with people at a party, and then she uploaded the pictures to an OSN like Facebook, making them available to everyone. Later, she tagged Bob and Carol who are co-workers (i.e. added hyperlinks to indicate Bob and Carol). They have many common friends and co-workers with Alice, and might find the pictures particularly shaming. Existing access control mechanisms in most developed OSNs place the burden of regulating policies over who can access the shared data solely on the owner of the profile where the data is. In our example, the privacy setting of the photo is only specified by

Alice who is the uploader and owns the photo in her profile, regardless of the privacy requirements of the users who are explicitly recognized through tags. One study shows that 75% of their participants were aware of the photos that have been posted of them by other people via email when being tagged in a photo [Facebook, et al. 2013]. So, even if they are aware of the fact that their picture is displayed and controlled by other users, they cannot impact the privacy preferences applied to this photo. Although a tagged user can detag her/himself to remove the explicit hyperlinks, the photo still exists in the OSN site and the user cannot stop other tagged users from sharing the photo with the people they have relationships with.

While OSNs are clearly considered to be a collaborative environment where the majority of activities involve at least two parties, current access control mechanisms and authorization administration suffer from collaborative policy limitations. This lack of collaborative policies for access control and co-administration violates the privacy of all stakeholders who share a particular content with the uploader by letting her/him take full responsibility over their privacy settings, which is extremely ineffective. Designing a suitable approach to address such a problem is the objective of this thesis.

Our work can be seen as a new step towards an access control model for OSNs. We first introduce and analyze scenarios, where more than one user should be involved in the process of making a collective access control policy. Those analyses and determinations are critical to the success of having collective privacy management of shared contents. To make our explanation of sharing patterns easier, we classify them into three types: profile sharing where accessors are the social applications, relationship sharing where a relationship between two users denotes a shared item, and content sharing which is the main type of sharing pattern and has the most sub-categories. Additionally, we precisely investigate all cases and subcases of content sharing patterns and define all users who have the right to participate in the process of making a collective access control policy. Then, we propose an approach to enable multiple controllers of the shared item to collaboratively specify the privacy setting. We begin with a formulation of the model and privacy policy specification and the result of this phase is an access control policy, *P*, from each associated controller. Often, multiple users have diverse privacy requirements over shared content; hence as part of building a collaborative access control model we propose a strategy to resolve the conflicts. Our principles for the conflict resolution strategy are essentially chosen to achieve the desired equilibrium between the privacy protection of online users and the value of sharing in OSNs' sites. Next, by taking into account those principles, policies that are individually regulated by each associated controller and our multiple controllers' scenarios, we develop a flexible and lightweight framework to achieve the collaborative privacy polices governing who can access and share the shared items in OSNs.

Our proposed approach includes three algorithms to address the problem of collaborative privacy polices .The first algorithm, called PermittedandDeniedAccessors, produces the final lists of accessors who are permitted to view the shared data and those who are denied. Based on our evaluation and determination for relationships between controllers and shared items and the type of activates applied to items, we have seen that dividing accessors into viewers and disseminators is an effective security practice. Moreover, according to our investigation of shared data situations, shared data dissemination comes in two varieties. Consequently, we introduce the AccessorSharing and ControllerSharing algorithms to reach collaborative decisions about who can disseminate the shared item. The AccessorSharing algorithm produces list of accessors, who are allowed to disseminate the shared item with their social networks (e.g., friends, family members, classmates, etc.). Because we have several associated controllers and they can disseminate the shared item with users who originally could be unauthorized to access the shared item, we come up with a ControllerSharing algorithm. We have formalized the ControllerSharing algorithm to enable controllers to regulate their privacy and protect their items from being used against them in some way.

To demonstrate the efficacy of the approach, we have implemented our PermittedandDeniedAccessors algorithm. A proof-of- concept prototype is to show the usability and feasibility of such an approach to achieve a collaborative access control policy over who can access a shared item. Our prototype application simulates all multiple controllers' scenarios, where contents are related with multiple users who are explicitly identified through posts, shares, tags or other metadata, as a first step. Moreover, our prototype application enables each associated controller to specify her/his privacy policy. Then, we run PermittedandDeniedAccessors algorithm to collaboratively produce the final lists of accessors who are permitted to view the shared data and those who are denied. In order to validate our approach, we conduct experiments using our prototype and discuss the results obtained in detail.

The organization of the remainder of this dissertation is as follows. We begin by reviewing the fundamental, relevant concepts in Chapter 2. It covers the four dominant themes of the research that are online social networks (OSNs), which is the platform of our problem, and several privacy issues and concerns in OSNs. In Chapter 2, we also review the access control models of data management systems and investigate and discuss the crucial requirements that an access control model for social network services should have. We end this chapter by presenting the related works that propose access control solutions for OSNs. In Chapter 3, we analyze and explore a number of OSN's scenarios where items are linked to numerous users who are explicitly recognized and have the right to participate in the shared item's privacy setting. Afterward, in Chapter 4, we formulate an access control model that determines the essence of the collaborative authorization requirement. Furthermore, we present a collaborative policy specification scheme for access control and authorization administration. We also explain the principles of our conflict resolution strategy. Finally, Chapter 4 provides algorithms that we design to address the problem of collaborative privacy polices for shared items in OSNs. Chapter 5 presents the implementation details of a proof-of- concept prototype of our approach to show the usability and feasibility of such an approach. We conclude the dissertation and outline future research directions in Chapter 6.

Chapter 2

2 Literature Review

This literature review explores the four dominant themes of the research. First, we offer a brief overview of online social networks, which is the platform of our problem. Second, we discuss some privacy issues and concerns in OSNs that we address in this thesis. Understanding these risks and challenges helps to design a suitable approach to address them. In the next section, we provide overview of access control models in data management systems (DMS). Furthermore, we investigate and discuss the critical access control challenges in OSNs; then, we present the key requirements that an access control model for social network services should have. This section ends with review the main access control solutions for OSN. We categorize existing related works into two types based on type of administration policies. We discuss access control models for OSNs that do not consider collaborative authorization administration of shared data in OSNs. Furthermore, we review and discuss in detail the approaches that have been proposed to provide a collaborative policy for OSN access control models. Finally, we provide a brief discussion of the trust notions in social network because we combine trust values between individuals in our algorithms.

2.1 Online Social Networks

Networks have been used in many systems such as computer networks, the World Wide Web, biochemical networks and social networks. Each of these networks consists of a set of nodes or actors representing, for instance, web pages on the World Wide Web, connected together by edges or relations, representing links between web pages. Our focus in this thesis is on social networks on the Internet. However, we start by describing some basic ideas related to social networks in general. Primarily, a social network can be compared to the concept of society where individual actors (nodes) are connected with each other by relationships (edges). Thus, actors and relations represent the building blocks of social networks; first of all, an actor is a social entity that socializes with other entities to maintain existing relations or to establish new ones [Wasserman and Faust.

1997]. Furthermore, a relation symbolizes a linking between two actors. On social networks, the concept of relationship is important when studying the structure of social networks, and is described by numerous features such as its type, direction, strength, and weight.

Since the World Wide Web and the Internet are continuously increasing in their popularity, massive types of services are available through them. In this context, a virtual community has been created for users to interact and collaborate with each other on the Internet, known as an online social network (OSN) [Chiu, et al. 2008, Howard. 2008]. They can also be called social network sites [Boyd and Ellison. 2007], social web sites [Kim, et al. 2010], or online networking sites [Gross and Acquisti. 2005]. From this moment on, and in the rest of the thesis, for simplicity purposes, we refer to these web services using the simple and extensive term of online social networks (OSNs). An OSN is “a web based site that allows individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system. The nature and nomenclature of these connections may vary from site to site”, as said by Boyd and Ellison [Boyd and Ellison. 2007]. As the area of OSNs has become a development part of the online activities on the web and various online services insert constantly new social features to their offerings, the classification of OSN services broadens. OSN services range from social interaction-focused sites such as Facebook, Friendster, or MySpace, to data dissemination-centered services as Twitter, to professional expertise and accomplishments -centric networks such as LinkedIn, to social communication features added to existing sites and services such as Flickr, YouTube, or Amazon [Pallis, et al. 2011, MusiaÅ, and Kazienko. 2013]. These services are generated and maintained by commercial companies. Although each of these services has different characteristics of social interaction, using the previous definition, let us try to generally specify some basic properties of an OSN’s service. First, it is an online site, platform, or service that provides environment and tools to build the social network or social relations among people. Furthermore, OSNs offer a wide range of tools for their users (e.g. tagging, uploading, commenting, re-sharing, etc.) to share interests, activities, backgrounds, or real-life connections. Also, in OSNs, each user is defined by a virtual representation, often a profile, that contains the list of her/his connections (e.g. friends, colleagues, family members) and a variety of additional services; for example, they can participate in group or community activities and receive notifications on the activities of her/ his connections.

For the sake of clarity, we briefly explain the concepts of social network users, user profiles and social links precisely in the context of OSNs. Firstly, the user is the core of an OSN; the majority of existing definitions of social networks are centered around users [Boyd and Ellison. 2007, Schneider, et al. 2009]. When these users join social networking sites, they have to create a personal profile to upload and post their items such as photos, opinions, etc. Then, the user makes connections to entities/individuals already members on these sites. Also, she/he socializes by interacting with other users by a wide variety of communication tools provided by these sites. Second, an OSN's user is virtually represented by a profile where identifying information (e.g. name, age, gender, photos and online status) and user's interests and preferences (e.g. joining groups, rating movies, books or music and liking brands) are shown. Consequently, user profiles shape users' personalities, identities, and behaviors on social networks you [Mislove, et al. 2010]. Finally, a user usually socializes with other network members who are already known to her/him. Additionally, a new member can establish new connections to individuals who are often suggested by the site upon the shared personal information such as name, location, photographs, birthday, personal interests, etc. Although users actually get linked to different types of contacts such as friends, family members, colleagues, co-workers, and strangers, current OSNs rarely discriminate between social relationship types. Indeed, OSNs often provide only a default relationship type that connects users of particular OSN site to each other. For example, social links are known as friends, regardless of its veracity or trust level.

Nowadays there are hundreds of OSN applications available in the World Wide Web, supporting a vast range of interests and practices. As a result, OSNs have become one of the highest used Web applications in our lives, so they dominate the time that users spend on the web [Goel, et al. 2012]. Many statistics indicate that OSNs have become a fundamental part of the online activities on the World Wide Web all over the world. A Nielsen study [Nielsen. 2012]shows that online users often are willing to spend their web time on social networks and blogs. According to a report by Socialbakers.com, which shows how much the number of OSNs users rapidly increases, Facebook has almost 1 billion users. Thus, currently it is considered to be the largest OSN with more or less 1 out of 7 people in the world having a Facebook account [Hampton, et al. 2011, Socialbakers. 2014]. Moreover, Twitter, which is under the scope of business and entertainment social networks, has 230 million active users (as of June 2014), tweeting an average of 500 million tweets every day [Twitter Inc. June 2014].

Due to the large size of OSNs, graphs are the most appropriate and common way to represent these networks. Consequently, the platform of OSNs is the social graph, where $\mathrm{G} = (\mathrm{V}, \mathrm{E})$, with vertices $V = \{v_1, \ldots, v_n\}$ corresponding to individuals and edges $E = \{(v_i , v_j)/ \; v_i \; v_j \in V, \; i \neq j \,, 1 \; \leq i, j \; \leq n\}$ modeling the relationships among vertices [Mislove, et al. 2007]. Under the umbrella of social networks are many different types of graphs to represent different forms of data and to pattern the structural properties of social networks. Graphs can have their edges directed or undirected, weighted or un-weighted, labeled or unlabeled. In OSNs, both directed and undirected graphs can be used depending on the nature of the relationship. For instance, Facebook is a typical example of social network formed as an undirected graph, where friendships between users are symmetric. In other words, R_{ij} or R_{ji} both indicate a friendship link between user i and user j. By contrast, Twitter is an example of a directed social network where relationships are not bidirectional. An edge between two nodes represents the followee-follower relationship, thus R_{ij} denotes user i is following user j.

Recently, we have been witnessing the popularity of a large variety of OSN sites increase rapidly. In fact, the main driving force behind their proliferation and success is that they offer ingenious ways for users to create, search and manage their own OSN communities. Additionally, they are usually open systems and a valuable source of social network data. Consequently, with this rapid expansion of OSN sites and the radical shift in the number of involved users around the world, there has been increasing concern about the privacy of individuals participating in them. The case of OSN is especially important because such open availability of personal information highly exposes OSNs users to a number of security and privacy risks. Researchers from different computer science disciplines have investigated some of the privacy problems that uniquely arise in

OSNs (e.g. [Chi Zhang, et al. 2010, Lewis, et al. 2008, Carminati, et al. 2014, Gail-Joon, et al. 2011, Hongyu Gao, et al. 2011, Mahmood. 2013]). Indeed, the privacy issue within the context of OSNs opens a new interesting challenge in the research community and has received a lot of attention.

2.2 Privacy Threats in Online Social Networks

Nowadays more and more information is publicly available in OSNs. These networks are increasingly attracting the attention of users who are interested in preserving existing relationships, making new connections, and using the numerous social networks' services for example, sharing photos, ideas, interests, events, and activities within their individual networks. Although massive amounts of private and personal information could be delivered through different Web sites such as medical information, taxes or social networks, the data delivered by OSNs is especially important because they have been gaining popularity among Internet users. Also, OSNs are expanding rapidly and providing more personal information than we could ever expect. For example, Google+ has 170 million active users (as of April 2012) and Pinterest, which is the third most popular online social network in the U.S., had collected 10.4 million users as of February 2012 [Google Official Blog. April 11, 2012, Experian Marketing Services. 2012]. Moreover, LinkedIn boasts on its web site that it has added more than 23 million members since December 31, so in April 2014, it has reached 300 million members in more than 200 countries and territories [Deep Nishar. April 18, 2014]. Although Facebook was only launched eight years ago, at the time of writing this thesis the total number of daily active users is closing in on 1 billion [Facebook. September 2014]. Hence, the availability of these vast amounts of information within OSNs obviously raises privacy and confidentiality issues.

With respect to other Web applications, OSNs with over 1 billion users present new challenges concerning the privacy of personal information. In fact, OSNs are built on interaction, where users often willingly share personally identifying data about themselves, and mostly they are open systems. Online users arguably have an unclear idea about who can access their private or semi-public information or what portion of their information needs to be accessed. While the risk of exposing personal information and the user population’s lack of awareness are addressed and described by several studies and recent news reports (e.g. [Carminati and Ferrari. 2008, Gates. 2007, Joshi and Kuo. 2011, Gurses and Diaz. 2013, Gail-Joon, et al. 2011, Hongyu Gao, et al. 2011, Mahmood. 2013, Raad and Chbeir. 2013, Loukides and Gkoulalas-Divanis. 2009, Gross and Acquisti. 2005, Becker. 2009]), OSNs’ security and privacy requirements still are not well understood or completely expressed. Indeed, there are explicit differences between the privacy issues in databases and in OSNs, especially in the way they are shaped, maintained, and in their use. Classic databases are created by an entity with the objective of keeping track of individuals according to diverse principles and with a specific purpose. The major objective of database security is to provide data security from unauthorized access and use. In other words, individual entities whose information is stored in a database have no access to each other’s records. This privacy issue occurs when the owner of databases is asked for some data and needs to transfer some of the data to accomplish this request. On the contrary, in OSNs Internet users join voluntarily to share their personal information, which are represented by different types of data, with one another. It is clear that sharing personal information, news and other contents are the main purpose of OSNs. Therefore, OSNs allow Internet users to access other network user’s information, thus an attack on user privacy often comes from another user in the same social network. Though security and privacy issues arising in OSNs are different from those in databases, the definition and treatment of privacy in these social networks is inspired by the corresponding concepts and philosophies in databases.

In this section we discuss the main privacy threats to OSNs, then in the following section, we present the access control model as solution to address the existing security and privacy concerns and issues related to OSNs. When this is discussed in the context of OSNs, the word *privacy* is used. Traditionally, in the computing field, people have talked about *access control*. Privacy in databases often involves statistical inference of specific facts about people whose information is stored in a database. Access control in computing systems determines who can perform operations on (or just read) specific pieces of data, or files. Since most of the data stored in OSNs is discrete, i.e. facts about users’ backgrounds, date of birth, etc., or involves discrete items like a photograph or a post, in traditional computer security deciding who can perform operations on this data would be considered access control. However, since much of the information on OSNs deals with people, or their opinions of posts by reposting something or liking something, deciding who can access this data is usually referred to as a privacy issue. We will use both terms in what follows, and in the context of this work, both privacy and access control concerns refer to the same thing – who can see something on someone else's profile.

Researchers from different computer science disciplines have investigated some of the privacy and security problems that arise in OSNs (e.g. [Pesce, et al. 2012, Gurses and Diaz. 2013, Squicciarini, et al. 2009, Hongyu Gao, et al. 2011, Mahmood. 2013, Raad and Chbeir. 2013]). Data privacy is defined as "freedom from unauthorized intrusion"[Vaidya, et al. 2006]. However, what are the fundamental assumptions about an unauthorized intrusion in OSNs is an open question because the concern about the security and privacy of OSNs' users and resources is a young field.

Privacy on OSNs is a complex concept, which encompasses major challenges. Indeed, researchers working from different perspectives differ not only in what they define as privacy but also in their fundamental assumptions about what the privacy threats in OSNs are. However, several topics regarding privacy breaches in OSNs are attracting more and more interest among scholars such as the surveillance problem, attack technique, users' limitation, disclosure, design flaws, limitations, etc. In general, we organize these problems into two broad categories: social privacy and institutional privacy. *Social privacy* focuses on how and when personal information of online users is shared with other users because the users are the primary consumers and component of these services. Moreover, social privacy can be defined according to Gurses and Diaz [2013] as "problems emerge through the necessary renegotiation of boundaries as social interactions are mediated by OSN services". Privacy disclosure that could arise from users' lack of awareness, the services provided, third party apps, other users, or privacy setting flaws and limitations. In contrast to the social privacy perspective, institutional privacy refers to undesired access by governments, service providers, or corporations to OSN users' personal information and social interactions stored on the OSN company's servers. The surveillance problem usually occurs by using data mining techniques. Social privacy and institutional privacy in OSN applications are two macro areas including several privacy issues. Our objective is to discuss several problems that lead to our specific issue that we address in this thesis. Subsequently, because institutional privacy does not specifically belong to our problem, we have chosen to leave it out of this dissertation's scope.

Social privacy (disclosure) issues have become a major concern for both OSN site's users and providers. We distinguish the four types of social privacy problems where information acquisition is based on privacy disclosure: users lack awareness, design flaws and limitations, breaches from other users and collective privacy management limitations. These issues are not new by themselves, but they are unique to OSNs and are worth reexamining.

- Users limitation:

User awareness is an aspect that the literature has repeatedly emphasized (e.g. [Ngeno, et al. 2010, Becker. 2009, Li, et al. 2013, Acquisti and Gross. 2006]). While revealing information on the web is a voluntary activity on the part of the users, most users are not aware of who is able to access their data and how their data can possibly be used and redisseminated. Vorakulpipat et al. find that while 52% of the respondents claimed to have a sufficient level of information privacy awareness, 75% of the users could not identify basic information systems security threats such as phishing, identity theft or attribute disclosure [Vorakulpipat, et al. 2011]. Due to the lack of awareness among Internet users, huge amounts of user's personal information such as pictures, contacts, and videos, are quickly falling into the hands of authorities, strangers, adversaries, recruiters and the public at large [Aimeur, et al. 2009]. Indeed, users' lack of awareness can lead to identity and attribute disclosure, which arises when an unauthorized user is able to access and determine the value of a sensitive user attribute, one that the user intended to be available only for authorized users. For example, Congressman Wiener shared his inappropriate pictures, occasioning public controversy leading to his resignation [Barret and Saul. 2011]. Furthermore, the earliest studies on concrete social network services present experimental evidence of the extraordinary self-disclosure practices within the sites. Works by Stutzman [2006], Lampe et al. [2006]and Acquisti and Gross [2006]find that students in the different university networks vastly shared sensitive information on Facebook. Despite this, it seems nowadays that Internet users are becoming more and more aware of privacy risks connected to OSNs. In fact, for the majority of privacy threats, if users do not take the initiative to protect their data, most defense mechanisms would fail disastrously.

- Breaches from other users:

One of the main types of attacks on OSNs is attacks from other users. OSNs offer Internet users new and interesting means to interact, communicate, and socialize. As a result, a huge amount of information, which could contain political views, a link they want to share, thoughts, sexual orientation, etc., expose a lot about the user. So, she/he will be of interest to various groups including friends, friends of friends and strangers to attack. All major OSNs allow a user's friends to view the personal information the user has uploaded to her/his space by default, but prevent others from doing so. However, in OSNs the notion of friendship is merely a social link that the two users have arranged to establish in that OSN, regardless of their actual offline relationship. Consequently, this contradiction causes a possible stealing of personal information by authorized users in OSNs. Several studies present potential attacks and risks from friends in OSNs [Mahmood and Desmedt. 2012, Mislove, et al. 2010, Akcora, et al. 2012]. In [Mahmood and Desmedt. 2012]the authors familiarize the targeted friend attack by creating a pseudonymous profile. Their single pseudonymous profile had access to the private information of 4,339 users. Beyond user-to-user relationships, most OSNs offer friend recommending systems to recommend a list of other users whom this user may know, have mutual friends or have common attributes. This feature leads to social link disclosure, which happens when an adversary is able to find out about the existence of a relationship between two users. Furthermore, a social link disclosure could lead to a cross-site profile cloning attack. Work by Jin, et al.[2013] demonstrates that an attacker using just one attacker node (2-hops) can identify more than 60% of a user's friends. Indeed, threats from the mutual friend feature are potentially more dangerous than breaches from friends because it's less likely to arouse suspicion. On many occasions, we accept a lot of friendship requests on OSNs without thinking about the short-term or long-term consequences of such a relationship.

- Design flaws and limitations:

Regrettably, current OSN applications indirectly necessitate the users to become system and policy administrators and experts to protect their online information. Another major problem that causes personal information disclosure is that users face great difficulties and complexities in effectively configuring their privacy settings. Furthermore, the risk from limitations and flaws of privacy setting design is greatly increased by OSNs' rapid growth as well as their continual adoption of new services. The design limitations and flaws refer to the weak privacy controls offered by current OSNs and the possibility of explicit attacks. One of the major design flaws that might abuse or misuse such critical and sensitive information is the default or 'recommended' privacy setting. While social network sites provide five different granularities for their users: only me, specific friends, friends only, friends of friends, and everyone, the default setting for most pieces of content is public (everyone) meaning the user shares her/his content with all one billion Facebook users if they do not change or modify their privacy settings. The results of Liu, et al.'s [2011]work reveals that 36 % of Facebook items still remain shared with the default privacy settings, general. Only 37 % of users have expectations which correspond to the default privacy settings; thus when these are incorrect, pieces of information are exposed to more users than expected. Another design weakness is the limitation in the granularity of photo privacy settings. Pictures are categorized into albums, and privacy settings are specified on album granularity, hence all pictures in a particular album have the same privacy preference.

To successfully use OSNs' privacy settings, users must first find them and understand their terminology. Additionally, they need to understand the privacy-related consequences of their behaviors and decisions. However, privacy policies in existing OSNs are too complex for most ordinary users to manipulate and understand [Johnson, et al. 2012, Raad and Chbeir. 2013, Anderson and Stajano. 2013]. In fact, most users have no experience to articulate their privacy preferences effectively. Further complicating this issue is the frequent changing these settings have over time, so the complexity of elucidating privacy settings is greatly increased. Also, the privacy tools are not meaningful enough to express users' disclosure preferences. According to a recent study in[Madden. 2012], 48 % of OSNs users still face some level of difficulty in controlling the privacy settings on their account. In all, current OSNs privacy mechanisms skip significant metrics such as risk, trust, humanization design, and some social metrics.

- Collective privacy management limitations:

In spite of the fact that collaboration and sharing represent the main building blocks of existing OSN sites, OSNs yet do not support any mechanism for collaborative management of privacy settings for shared data. While owning photos and posting comments to users' profiles are considered as the general purpose of social network services, users can be tagged in photos and reply (comment) to or re-share an existing post. Since multiple controllers' scenarios are raised by using different tools in OSNs such as posting, tagging and re-sharing, collaborative administration policies are becoming more and more important in many OSNs scenarios. Let us discuses the activities of photo sharing and tagging that are an integral and exceedingly popular part of profiles on most OSNs like Facebook. For example, Facebook hosts 250 billion photos (as of September 2013) [Facebook, et al. 2013]. On an average day, more than 350 million photos are uploaded. Besides photo sharing, tagging represents one of the prominent features of OSNs; it has often become part of personal identity management[Besmer and Richter Lipford. 2010], instead of merely sharing activities with friends. The first limitation of photo sharing activity in current OSNs is that when uploading photos, the request for permissions from other users appearing in the photo is not demanded, even if they are explicitly recognized through tags or other metadata. In[Facebook, et al. 2013], researchers show that being tagged in shared photos is the most prevalent way that participants know about photos they were depicted in. 75% of their participants were aware of the photos that have been posted of them by other people via email when being tagged in a photo. Consequently, the tagging feature enables internet users to review all photos of them that exist, which leads us to the hypothesis that being tagged in shared photos is seen to have some privacy benefits. However, existing access control mechanisms in OSNs choose to place the burden of privacy policy solely on the owner who has shared data in her/his profile, raising serious privacy concerns. For example, suppose Alice took pictures with people at a party, and then she uploaded the pictures to an OSN like Facebook, making them available to everyone. Afterward, she tagged Bob who is one of the people in the pictures (i.e. add hyperlinks to indicate Bob). He shares many friends and colleagues with Alice, and might find the pictures particularly shaming. Indeed, existing access control mechanisms in OSNs offer individual (rather than group) processes to regulate the decisions of accessing shared data. In our example, pictures are solely controlled and managed by Alice who has them in her profile, even though he is aware of the fact that their pictures are displayed and controlled by other users, Bob has no control over them and cannot impact the privacy preferences applied to these photos. Even if tagged users can detag themselves to remove the explicit hyperlinks from the photo to their profile, the photo still exists in the social network site. Also, they cannot stop other tagged users from sharing the photo in their social networks. As a result, current access control mechanisms and authorization administration suffer from collaborative policy limitations. In fact, shared pictures have an impact on the privacy of all users who appear in a picture and are mentioned in metadata like tags and comments. Henne, et al. [2014] report on how well their participants feel informed about two dissimilar types of photos of themselves on social network sites. 56% of their participants claimed that their level of information about acceptable photos is worse than neutral. In contrast, in the case of inappropriate photos, 70% of them selected a level worse than neutral to totally unsatisfactory. Therefore, current OSN privacy mechanisms violate the privacy of all stakeholders who share a particular photo with the uploader by letting her/him take full responsibility over their privacy settings, which is extremely ineffective.

The significant privacy threat of information disclosure is increased by limited and poor access control mechanisms for shared data in OSNs. They are clearly considered to be a collaborative environment; the majority of OSN scenarios involve at least two parts. Designing a suitable approach to address this problem is the objective of this thesis. In this regard, more insight into scenarios where contents are associated with multiple users who are explicitly identified through posts, shares, tags or other metadata is presented in Chapter 3. Finally, in spite of the fact that a lack of a joint administration policy raises a number of important issues (e.g. [Li, et al. 2013, Vorakulpipat, et al. 2011, Acquisti and Gross. 2006]), this problem has only been explored in few studies [Squicciarini, et al. 2010, Carminati and Ferrari. 2011, Sun, et al. 2012, Hu, et al. 2013, Xiao and Tan. 2012], which we will review in Section 2.3.2.2.2 .

Although these privacy issues are addressed and discussed as if they are independent phenomena, they are closely intertwined and are sometimes combined. For instance, because users are generally unaware about who can access their profile and specifically which parts of their profiles are accessible and to whom precisely, other users can easily violate these users' privacy. Moreover, the issues of design flaws and limitations, and users' lack of awareness differ in the way they are tackled, but they are explicitly and implicitly entangled. For example, users still create threats to their own privacy by using inappropriate and limited access control settings. Similarly, the problem of poor access control for shared content is not independent of the attacks from other users problems. The lack of a collaborative administration policy might have consequences for the effectiveness of maintaining the associated users' privacy boundaries by not allowing them to participate in shared data privacy settings. Indeed, all these issues refer to major gaps and limits in the architecture of access control mechanisms.

Finally, note that all of these privacy issues indicate that the existing primitive access control mechanisms as well as their designs must be improved to better address threats and meet users' expectations. With respect to traditional environments, in online social networks the increased risk to personal data processed is highlighted because this information is connected to user profiles, furthermore spreads across users' social activities and communications. Consequently, to address the privacy concerns, we believe improving the OSN access control schemes seems to be the first step toward addressing the existing security and privacy concerns related to online social networks. Nowadays, conventional access control techniques have several disadvantages when it comes to protecting the privacy of online social network users. Hence, the need for new access control mechanisms that are integrated with privacy preserving techniques specifically tailored to OSNs and based on metrics such as trust, co-controllers, and social metrics is becoming more compelling.

2.3 Access Control Models

As we discussed above, recently we have seen unprecedented growth in the popularity of OSNs for sharing data such as pictures, videos, audio, etc. between people. It comes with scientifically challenging problems and concerns about protecting the information of online users. In OSNs, users are encouraged to broaden their social network and to share their content with others; consequently, those activities obviously raise user's privacy and content confidentiality issues. Online users usually want to selectively share their content in OSNs; thus, an access control mechanism is what gives users more control on the spread of their information. Access control techniques are intuitively introduced to protect users' privacy, prevent unauthorized access and to selectively share contents in social networks. While access control mechanisms are simple and may only require comparison of credentials, OSNs have specific access control requirements, due to their particular characteristics. They require more fine-grained control and new controlled data sharing solutions that respect what has been proposed so far in the field of database management systems (DBMSs) and operating systems. Theoretical work on access control has led to several frameworks for achieving the desirable access control requirements for social networks and representing access control mechanisms in a flexible and fine-grained way, (e.g. [Carminati, et al. 2009a, Xiao and Tan. 2012, Hu, et al. 2013, Fong. 2011b, Cheng, et al. 2012a, Anwar, et al. 2010]. Nowadays, these access control frameworks, which are proposed for OSNs, replace a particular policy model such as discretionary access control (DAC), mandatory access control (MAC), or role-based access control (RBAC) paradigms, which are used in traditional systems. DAC is a means of restricting the access of subjects to objects based on the identity of subjects and a set of authorizations. DAC provides discretion to individual users over who is allowed to access the data they create or own. Also, in this type of access control pattern, a user has comprehensive control over all system resources that are owned and executed by her/him. In contrast, MAC utilizes security classification labels that represent security domains. Thus, the subject's and object's security classification determines the accesses that subjects can execute on the objects in the system [Ferrari and Thuraisingham. 2000]. The security classes associated with subjects are a measure of how trustworthy the subjects are; so, this trust model is based on subject labels defined as a clearance level [Ferrari. 2010]. The sensitivity of data is measured by object security levels that are enforced by the system. Consequently, in a MAC paradigm, user privileges cannot be passed from one user to another and there is no concept of ownership. In addition to DAC and MAC, in 1996 Sandhu et al. proposed the RBAC approach where restricting resources or system access to authorized objects is regulated according to their role (job descriptions) [Sandhu, et al. 1996]. In RBAC, privileges are associated with roles instead of being resource-owner centric as in DAC or security classification labels as in MAC.

In this section, we first briefly present the basic concepts of access control and give a brief history of the main research in the field of access control.

2.3.1 Access Control Models in Data Management Systems

Access control is one of the most important features of today's systems to protect data stored into Data Management System (DMS) [Ferrari. 2009, Bertino and Sandhu. 2005] . The technical ability to do something with a computer resource such as view, modify, create or delete is what we mean by access. Access control usually determines whether the ability is explicitly enabled or restricted in some way [Bertino and Sandhu. 2005]. Its overall goal is the protection of DMSs resources (i.e., data and services). Indeed, DMS access control focuses on addressing three main issues [Samarati and de Vimercati. 2001, Bertino and Sandhu. 2005]:

- Data confidentiality or secrecy refers to preventing improper or unauthorized access and to limiting data access and disclosure to authorized users.
- Data integrity that is protecting data from intentional or accidental unauthorized modifications or deletions by restricting the number of users with access
- Data availability ensures that authorized users have access to information resources. Also, it refers to preventing hardware and software errors that could make some of the data unavailable to authorized users.

Under this conceptual umbrella of access control preservation and according to their functional purposes, DMS access control mechanisms are divided into several categories such as logical or technical, compensation, preventive, administration, recovery, corrective and detective. Access control mechanisms are a central element of computer security, since they are the basis of implementing both information confidentiality and integrity. ITU-T Recommendation X.800 defines access control as follows:

"The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner."

Access control mechanisms are used to enforce a policy, which is specified by Security Administrators (SAs) or users, of restricting access to a data object to only those users (subjects) who are authorized [ISO. 1989].

2.3.1.1 Basic components of access control

According to the aforementioned definitions of access control, the basic concepts of access control are access control policies and authorizations. The policies are considered as inputs for access control and as high-level rules according to which access control must be regulated. Indeed, policies, which may be made by a management official responsible (e.g., security administrators, users, etc.) for particular systems, applications, or resources, articulate who can access which objects and in which manner (e.g., read, write, execute, delete, etc.), and, optionally, under what conditions (e.g., time, location, history, etc.) [Stallings. 2008, Benantar. 2006]. The rules may depend on many varied factors such as roles, permissions, dynamically deduced rights, or the specific user requirements. A simple example of access control polices is granting authorization to managers to read the psychological evaluations of employees only during their work shift. In order to enforce polices automatically, they should be translated into a set of authorizations. So, whenever a subject requests to access an object, the access control mechanism verifies the rights of subject against the set of identified authorizations.

The basic form to represent and store authorizations can be abstractly modeled as a triple (subject, object, access mode) indicating that the subject is approved to access the object under a specific access mode. While authorization can depend on the protected objects represented, there is a uniform way to represent the authorization; for example, in a relational database system, the authorizations regularly are stored in the system catalog as tuples. An access control model provides a formal representation of the authorizations and their enforcement. Then, by the access control mechanism that works as a reference monitor, a decision is made whether a subject's request can be granted totally or partially or should be denied. Indeed, the reference monitor is the basis of access control mechanisms and the trusted computing base (TCB) component of a computing system. Its concept, presented by Lampson in the early 1970s [Lampson. 1974], defines a set of design requirements on a reference validation mechanism that intercepts every access request (e.g., read, write, etc.) from a subject to perform on an object to determine whether a subject can be partially or totally granted or it must be denied [Benantar. 2006]. Thus, reference monitors have two main aspects: access control enforcement, and the computation of an access control decision. The development of the reference monitor must have some essential properties such as: it must unbypassable, that is, it must mediate all access requests to the systems and their resources. The second property is that it must be tamper-proof, to protect the reference validation from any alteration that could result from a malicious user. In the worst case, we must be sure the reference monitor is qualified to detect any improper modification. As we previously mentioned, authorizations are the second basic building component for access control that declares, in a basic format, who can access which object and under which mode or condition. However, this function may require further components to determine an access control decision that may not be accurately captured by the basic format. Consequently, extending the authorization format has been addressed by several researchers to offer more an inclusive authorization language [Jajodia, et al. 2001, Bonatti and Olmedilla. 2007, Bonatti, et al. 2009, Bertino, et al. 2000].

2.3.1.2 Administration of access controls

Additionally to the main two components of access control, administration is one of the most involved and challenging aspects and functions in access control. Access control administration deals with collection of duties, responsibilities and tasks that are appointed to administrators to grant or revoke authorizations. The access control administration function collectively involves monitoring, modifying, testing, and managing user accounts and accesses on the system. Administrative policies state who is allowed to grant and revoke privileges to subjects and regulate the performance of other administrative operations such as creation, modification and deletion of roles [Sandhu and Samarati. 1994]. Although this is one of the most significant aspects of access control, it is probably the least understood and most complicated aspect of access control. In order to understand access control administration, we introduce the three basic approaches to administering access controls that are centralized, decentralized or a combination of these, which often is adopted by hybrid environments. In fact, the appropriate decision to choose an access control administration method depends on the needs and circumstances of a particular organization and the sensitivity of its information [Ferrari. 2010, Guttman and Roback. 1995, Stallings. 2008, Sandhu and Samarati. 1994]. In a centralized administration approach, only one (or few) trusted entities is responsible for configuring access controls. The security administrator(s) is typically the central office where all access control design is done. Using centralized administration is very simple because policies are maintained and modified only in one central location. However, it is a very strict approach because the ability to grant or revoke the authorizations resides with very few individuals. Also, one of the main disadvantages in a centralized administration model is that if changes need to be done quickly, going through single authority location can be time consuming. The second administration mode is decentralized administration, which captures the simplicity and flexibility requirements of the real world. In this type of administration, the creator of an object becomes its manger that directly controls the access on the object. Moreover, a decentralized paradigm of administration is easier to implement because the access control unit is not a single point of failure. Although decentralized administration gives the users who are close to the resources the right to control the access, owners may practice security and access control in different ways that introduces conflicts. In addition, when access is not administered centrally, keeping all owners on track to regulate who can access their objects makes a general administration and invalidation of authorizations more difficult.

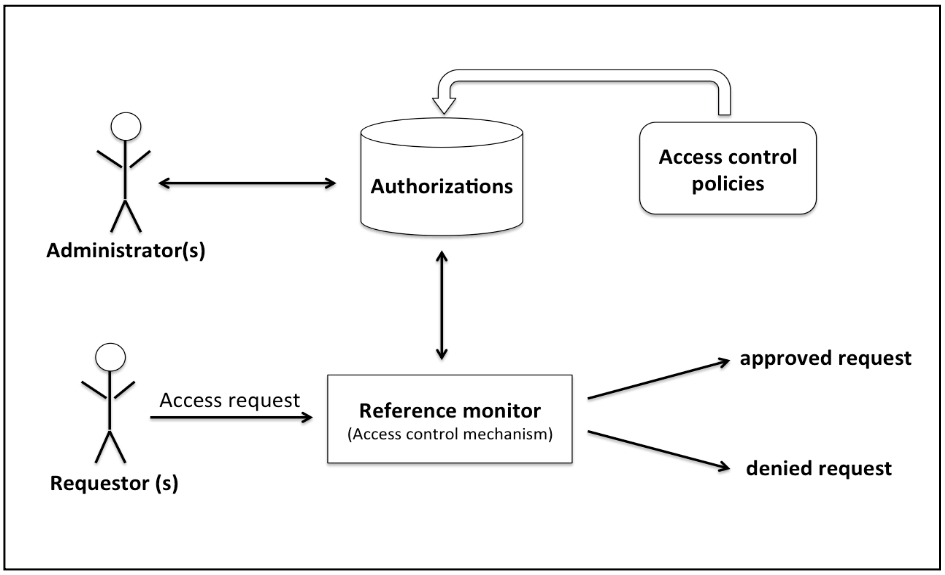


Figure 1: The main components of access control and their interactions.

The main components of access control and their interactions and coexistences between each other and with other security services are illustrated in Figure 1.

2.3.1.3 Access control paradigms

Many real world domains such as government, the military, enterprises, etc. have complex policies and various privacy anxieties, where evaluating access requests is based on the application of different requirements and rules. Access control techniques dictate how the administrator grants an access permission to a subject to perform her/his duties on a particular object. Access control models are frameworks for implementing components, deploying mechanisms and ensuring the integrity of security policies that facilitate authorized access to protected resources. There are several methods of limiting access to authorized subjects that have been devoted to improving access control. Three main types of access control models have emerged:

Mandatory Access Control (MAC):

The policies of mandatory control access (MAC) are based on mandated regulations determined by a central authority that is a security policy administrator [61]. Loosely, we can name any access control model that enforces security policies independent of user operations as mandatory access control. Thus, in MAC, the end user has no ability to override the policies or provide any privileges to any subject. Indeed, the concept of owner administration does not exist in MAC; however, its policies regulate accesses to data by subjects based on policies that are defined by prearranged subject attributes (e.g. security clearances) and object security levels that are enforced by the system [Brand. 1985]. MAC is associated with two security models: the first is the Bell-LaPadula model, which was published in 1973 and focused on the confidentiality (secrecy) of data[Bell and La Padula. 1976], whereas the second associated model, the Biba model is focused on the integrity of information [Biba. 1977]. MAC models are used as a response to the security requirements of the military and other governmental organizations where the system administrator takes charge of the full access control and keeping secrets is the primary goal.

Discretionary Access Control (DAC):

MAC, while highly important to military applications and governmental organizations, is not the most commonly used method of access control in commercial applications or operating system such as Unix, Linux and Windows 2000. These systems allow subjects, who could be the owner or creator of an object, to decide access rights on their objects. This model is called discretionary access control (DAC), and first appeared to implement Access Control Matrices introduced by Lampson in his paper on system protection[Lampson. 1974]. Indeed, this pattern is labeled discretionary access control or DAC because it is at the discretion of subjects to make policy decisions, which determine who can access the objects they create or own, and assigns security attributes such as accessors'/or requestors' identity and authorization rules[Brand. 1985]. Unlike the MAC framework where access cannot be passed from one subject (user) to another, in a DAC-based system, privileges can be transferred, suspended, resumed or revoked with relative ease [Benantar. 2006]. DACs are typically implemented in one of two ways: Capability-based security or Access Control Lists (ACLs). First, an ACL defines a set of subjects (users), which for example, may be listed as owner, group, or others, who are allowed to access a particular data object along with types of actions they can perform by using primitive control values such as read, write, execute, etc.[Ritchie and Thompson. 1983]. Second, capability-based security grants a subject the permissions with specific capabilities (e.g. has been issued to the requestor, hasn't been revoked or expired) to access the data object[Levy. 1984]. Furthermore, DAC can include and implement transaction controls, time-based controls and other fundamental systems of identity-based access control (IBAC). DAC permits a wide and common range of administrative policies, which represent important aspects of an access control model in general and discretionary access control in particular. Centralized, cooperative, ownership, hierarchical and decentralized are some of administrative policy approaches supported by DAC[Ferrari. 2010]. The bottom line is that using DAC provides flexibility and enabling of fine-grained control over system objects. These are the reasons why many application environments, especially commercial DBMSs, adopt DAC.

Role-Based Access Control (RBAC):

The need to specify and enforce enterprise-specific security policies in a way that satisfies and covers the requirements of most commercial enterprises and to simplify authorization administration have been the motivation for many researchers and practitioners to develop a new access control approach which captures real-world access control requirements. Hence, the role-based access control model (RBAC) has been invented for large enterprise solutions where a role represents a particular function within an organization and can be seen as a set of responsibilities that the user in this role can perform. Like a position in the real world, an RBAC role connects to a set of permissions to perform some operations or duties on some objects. The early references of using roles can be found in Baldwin's paper [Baldwin. 1990] where they are called protection domains. Then, in the early and middle of 1990s RBAC was introduced by Ferraiolo and Kuhn [Ferraiolo and Kuhn. 2009], Sandhu, et al. [1996] and Nyanchama and Osborn [1994, 1999], as a way to simplify authorization administration within companies and organizations. The standard for RBAC was proposed by Ferraiolo, et al. [2001]as one of the new generation of access control mechanisms.

In role-based security systems, the role is an intermediate element between subject (user) and permission. A permission denotes the ability that a subject (user) can execute certain operations on certain objects. Users are then made members of roles, thus acquiring the roles' permissions. Consequently, all authorizations are granted to the role associated with activities that can be executed on objects by getting these authorizations.

In contrast to Access Control Matrices (AM) and Access Control Lists (ACL) a user, under an RBAC model, is not directly authorized to any right, but has to activate a role that has already been assigned to her/him. Indeed, roles raise the access control to a coarse-grained level to manage accesses, so a user can be assigned to more than one role, which can be activated on different occasions. Moreover, different users can simultaneously play the same role. On the other hand, the role concept puts access control in a more granular manner that can specify what types of activities can be performed within an object, not just on the object as a whole. Furthermore, Sandhu, et al. in[1996] presented the family of RBAC models which includes sessions as a supplementary feature. A session represents what a user has activated at run time, and distinguishes RBAC from traditional group mechanisms. As a result, by session management we can easily restrict a user to not activate conflicting roles at the same time. The RBAC standard was proposed by National Institute of Standards and Technology (NIST) [Ferraiolo, et al. 2001], and is now an ANSI standard [INCITS. 2004]. Most studies confirm that roles are a useful and suitable approach for many commercial enterprises and government organizations. First of all, because roles signify a specific function within an organization and the number of roles of related authorizations is usually much fewer than the number of users or individual permissions, knowing what a user's organizational accountabilities are is easier to manage than assigning individual permissions to single users. Moreover, roles can also be simply used to manage and control the administrative mechanisms because the concept of role is more stable than individual user ownership in a large number of business activities. For instance, when user changes or deletes her/his function within organization, the security administrator only needs to cancel or change the roles corresponding to the user's job; when a new user joins the organization, the administrator simply needs to grant the appropriate role membership(s). Therefore, these transactions do not have any influence on the roles and their relevant permissions. Lastly, because of its relevance, RBAC has been extended into various models with the aspiration to address the access control challenges and satisfy the requirements of applications, commercial enterprises and governments such as Temporal RBAC (TRBAC) [Bertino, et al. 2001], Generalized TRBAC (GTRBAC) [Joshi, et al. 2005], Generalized RBAC (GRBAC) [Moyer and Abamad. 2001] , etc.

2.3.2 Access Control Models in Online Social Networks

In this section, we discuss the requirements for access control in OSNs. However, we will present the privacy settings that are implemented and used in current OSNs. Most OSNs provide only the most basic access control policies. For example, most social networks (e.g. Facebook, Google+, MySpace, Friendster, etc.) focus on profile privacy rather than settings for contents. Also, even though a few of them support grouping of users, the process of grouping users into lists as in Facebook or into circles as in Google+ is troublesome and time-consuming task for the end-user. Although access control in most current OSNs is essentially based on relationships among users and resources, the user is offered a limited number of privacy setting options by choosing options such as public, private, set of users whom she/he has a direct relationship or simple alternatives to these basic settings such as “friend of friend”. By these limited options, users may grant access to unauthorized users; nevertheless, they are restricted and inflexible in signifying authorized users. Even though most OSNs provide fine-grained control on profile elements such as personal photos, status or other basic information, they only offer one binary type of relationship that is friend or not. Consequently, this limitation violates a significant security principle which is keeping consistency between online and offline social networks [Chi Zhang, et al. 2010]. Various studies expose the complex and unfriendly access control interfaces of OSNs (e.g., [Lewis, et al. 2008, Boyd and Hargittai. 2010, Johnson, et al. 2012, Gross and Acquisti. 2005]); moreover, in some OSNs, the access control interfaces are very hard to find. Gross and Acquisti in [2005]measure and analyze users’ behaviors towards privacy policy and their participation with available privacy policies in Facebook. Their study shows that only a small fraction of users were conscious of the availability of the privacy settings. As a result, the users have to be experts to control portions of their data in OSNs.

Access control in OSNs carries numerous unique features different from access control in traditional data management systems. Also, the privacy issues and concerns in OSNs differ from that in databases. Subsequently, the definition and treatment of privacy in OSNs is dissimilar to that in traditional databases. Many traditional approaches that maintain privacy in data management systems such as MAC, DAC, RBAC, etc. are not appropriate options to work with in an OSN context. As we previously discussed in mandatory and role-based access control (MAC and RBAC), policy is typically specified by the security administrator. Access control policy in DAC is defined by the resource owner. These ways of specifying policies are working properly if users know their accessors because they are able to put up a set of permissions to accurately grant the access only to intended subset of their friends. However, in OSNs scenarios where users mostly do not know a priori all their possible indirect accessors (friends), a traditional access control policy is not enough because in such situations users will have to specify a huge number of policies. Similarly to DAC, *owner-based administration* is adopted by OSNs. In OSNs the administrator of data is a user who may desire to regulate access to her/his resources and activities related to herself/himself, therefore user-specified policies should be the access model for OSNs. Other than the content owner, some associated users (e.g. users tagged in photos owned or uploaded by another user or parent) may also desire to participate in the content's privacy setting to regulate how the content or user can be exposed. As we previously mentioned, a reference monitor is the fundamental component in access control, which scrutinizes each access request to the system based on the specified access control policies to determine whether the access request can be authorized or denied. Thus, to enforce the privacy policy in OSNs we need a suitable architecture for access control enforcement. Several studies have demonstrated semi-decentralized and fully decentralized solutions are more suitable to Web-based Social Network (WBSN) than a traditional, centralized approach (e.g., [Yeung, et al. 2009, Carminati, et al. 2009b]).

2.3.2.1 Access control model requirements for OSNs

OSNs are quite large and complex clusters of personal data and therefore we need new approaches to describe and execute access control on that data. Indeed, many recent research results demonstrating that users' actual privacy settings do not match their sharing intentions are particularly troubling[Madejski, et al. 2012, Liu, et al. 2011]. Access control in OSNs presents numerous unique characteristics different from access control in traditional data management systems. However, providing broad and ideal access control scheme requirements for OSNs has become a much more difficult task due to the increase in the amount of content shared and the increase in the number of users.

Furthermore, defining the structure of each social network and the nature of their data is not a simple task, because most OSNs combine different elements and more than one type of network (e.g., relationship network, trust network and group network). Also, OSNs may change their focus and provided services over time. In 2007, Gates identifies a set of requirements in order to successfully develop access control in Web 2.0, which includes OSNs [Gates. 2007]. Relationship-based, fine-grained, interoperable, data exposure minimization and sticky-policies are the necessary requirements to establish access control systems for Web 2.0, as identified by Gates [2007]. Truly, there is not a clear path that directs researchers and developers in the duty of developing an access control model for OSNs that tackles the limitations of the privacy settings and addresses the whole set of requirements. Although online social networks' security and privacy requests still are not comprehensively recognized or fully defined, some research has investigated requirements for access control systems for OSNs (e.g., [Carminati, et al. 2009a, Cheng, et al. 2012b, Cheng, et al. 2012a, Carminati, et al. 2014, Gates. 2007]). Based on Gates' [2007]considerations, and according to [Carminati, et al. 2009a, Cheng, et al. 2012b, Cheng, et al. 2012a, Carminati, et al. 2014], next we summarize the key requirements that a future access control model for OSNs should have.

- User-friendly: Providing a natural and user-friendly way of defining access control rules. Although social networking capabilities provided by Web 2.0 have increased, most of the access control interfaces are complex and not flexible. Some are very hard to find and use. An ideal access control model must be effective and flexible to use, matching real world scenarios.
- Flexibility with data: Social networks with different interests offer diverse access control mechanisms. However, an ideal access control model should have the ability to work with all types of data regardless of where they are stored.
- Fine-grained: It is essential to develop an access control paradigm for OSNs in a fine-grained format. Actually, most OSN applications focus on profile privacy, but online users should control their contents even for the shared data, selecting who is allowed to access it and under which conditions in a fine-grained method.

Also, allowing fine-grained control for data and accessors should be offered by a future access control model for OSNs.

- Relationship based: Recently, there is consensus that an access control model for OSNs should be relationship-based because it is very intuitive for users to adopt their social relationships to define authorized members for their information (e.g., [Masoumzadeh and Joshi. 2011, Fong, et al. 2009, Carminati, et al. 2009b, Carminati, et al. 2011, Fong and Siahaan. 2011, Cheng, et al. 2012b, Cheng, et al. 2012a, Hu, et al. 2013]). In OSNs, user-to-user (U2U) relationships between the accessing user and the resource owner is a common characteristic that relationships in OSNs have. Moreover, between users and resources there are some different types of relationships such as ownership, like, tag, comment, post, etc. However, existing OSNs treat ownership as the only manifestation of user-to-resource (U2R) relationship; consequently the authority of accessing resources is still controlled based on the relationships between the accessing user and the controlling user (U2U). Also, due to many functionalities presented by today's OSN applications and many user activities found in them, there exist several different types of resource-to-resource (R2R) relationships such as pictures under the same album, comments or likes to a blog post, etc. An ideal access control scheme must be able to express U2R and R2R relationships in addition to U2U relationships for authorization policies and decisions. Additionally, there are several characteristics that relationships in OSNs should have. First of all, types of relationships can be mutual such as friend or colleague and one directional such as parent-of or fans or followers. Furthermore, relationships can be direct (e.g., Alice has a direct relationship of type "colleague -of " with Bob) or indirect (e.g., Alice and Carol are not directly connected, but they are related in that Carol is a friend of Bob, and Bob is a friend of Alice). The type and depth of a relationship determine the level of information disclosure; for example, users commonly allow their close friends to view private information more than other types of contact. In contrast, it would be less embarrassing for users to share embarrassing photos with strangers than with close friends or even with friends of friends [Akcora, et al. 2012]. Also, relationship-based access control should take into account the specification and composition of complex relationships. Finally, in order to have more flexible relationship-based access control, users should be able to customize their relationships by defining names and properties.

- Interoperable: An access control model for OSNs should be able to exchange and make use of information between the multiple applications. Usually, users have accounts on different OSNs such as Facebook, Blogger, LinkedIn, etc. Thus rather than users redefining the access control policies for each individual site, they might want their policies and preferences to follow them regardless of where their data are stored.
- Trust-based: To improve OSNs' access control systems, the concepts of trust and depth have to be considered. An access control policy for OSNs should make a user able to state how much she/he trusts other users. Trust may be useful in determining who is authorized to access OSN resources because trust values are often associated with different levels of information disclosure[Jøsang, et al. 2007]. For example, users usually share confidential content only with the most trustworthy friends rather than with users who have the minimum trust level of friendship. However in some cases, users are willing to disclose particular data to anonymous strangers, but not to those who know them better such as family members or close friends[Joshi and Kuo. 2011]. In either case, trust values can be helpful to build an access control scheme that is intuitive and simple to use, keeping consistency between the way of managing online and offline social networks.
- Co-ownerships: Another key parameter that is essential to develop an effective and flexible access control manner for OSNs is joint administration of policy. For instance, consider the case where Alice uploads a photo in her profile, and assume that she tags Bob and Carol in the photo. Users tagged in a photo owned by another user may have some rights to control the release of the photo to other users. Indeed, due to the increase in the amount of content shared, collaborative policies for access control and authorization administration is becoming more and more important in OSN scenarios. An ideal access control model must consider all subject-to-object relationships, besides the traditional ownership

administration policy adopted by DMSs as we formerly mentioned.

2.3.2.2 Prior Access Control Models for Online Social Networks

Parallel to the increase of popularity, research on access control has been growing with the results of the proposal of several access control models. Although there is no a clear and well defined path for developing access control systems in OSNs, several proposed models attempt to address some of the requirements specified by Gates [2007]. In what follows, we discuss the main existing works addressing access control for OSNs. We categorize them into two types based on type of administration policies. First, we review approaches that do not consider collaborative authorization administration of shared data in OSNs. On the other hand, we review and discuss in detail the approaches that have been proposed with respect of the need for a collaborative policy for OSN access control models.

2.3.2.2.1 Access Control Models with Single Specification (Owner)

Improving and proposing a number of access control mechanisms for OSNs appear as the essential step toward addressing the existing security and privacy concerns of online users. In this section we provide an overview of existing research in the field of access control in OSNs. In fact, access control in OSNs presents a number of unique characteristics different from traditional access control techniques. In mandatory and role-based access control, the policy is typically regulated by the security administrator. However, in OSNs, users desire to specify policies to their resources and activities related to themselves; consequently access control in OSNs is subject to user-specified policies.

A number of proposals, in various levels of maturity, that attempt to develop usable and fine-grained access control mechanisms for protecting personal and shared data are emerging in OSNs. A first approach was proposed by Gollu, et al. [2007] , where a social-networking-based access control scheme suitable for online information sharing is given. Their approach considers users' identities as key pairs and identifies social relationships based on social attestations. To define and manage the access lists of users, they adopted simple access control lists. In [Hart, et al. 2007], the authors discuss the access control requirements of WBSNs in general, and OSNs in particular, and they proposed a content-based access control model. In this approach, relationship information available in OSNs is used to denote authorized subjects. However, resources are denoted by their content, which is derived based on content analysis techniques and users' tags. Hart, et al.[2007]solely considered direct relationships in WBSNs.

Initial access control models for social networks concentrate mainly on computing trust values for users, inspired by research developments in trust and reputation computation in social networks. Kruk, et al. proposed one of the earliest approaches that considers asymmetric friendships that are quantified in the context of Friend of a Friend (FOAF) in a distributed identity management system based on social networks [Kruk. 2004, Kruk, et al. 2006]. Also, they combine trust metrics and degree of separation policies to control accesses of friends to data in a social network. However, this approach supports only a one type of relationship and adopts centralized access control enforcement. For those reasons, this approach is not appropriate for the OSN domain, which is naturally dynamic and decentralized. Furthermore, to preserve the trustworthiness of users' data in social networks, Carminati, et al. propose a similar concept of trust-based access control model which is more mature [Carminati, et al. 2006]. They have proposed it in a semi-decentralized [Carminati, et al. 2009b] and in a decentralized architecture [Carminati, et al. 2008, Xue, et al. 2011] with relationship types, distributed trust metrics and degree of separation policies [Carminati, et al. 2006, Carminati, et al. 2007, Carminati and Ferrari. 2008, Carminati, et al. 2009b]. In the context of trust, Ali, et al. suggest to customize trust metrics to enforce access boundaries [Ali, et al. 2007]. This works by adopting a multi-level access control approach, where each user in an OSN has a security level that is computed as the average of the trust values assigned for her/him by other users in OSN. Also, all resources in OSN have security levels that are assigned by owners. Consequently, the access is based on the security level between user and resource, where if a resource's level is equal to or less than a user's level, the user is authorized to access. Moreover, automatically classifying nodes in regions is a different trust measure that is proposed by Villegas, et al. [2008].

However, these methods focus mainly on subject specification based on trust and distance measures. Carminati, et al. propose an approach that utilizes semantic web technologies [Carminati, et al. 2009a, Carminati, et al. 2011]. With the Web Ontology Language (OWL) and the Semantic Web Rule Language (SWRL) they define three kinds of policies, namely, access control policies, administration policies and filtering policies. Access control policies are permissions to access; administration policies are to specify who is allowed to define those policies; and filtering policies are used to limit someone's access to information by an administrator. This work offers a model that shows different user-user and user-resource relationships. Some others propose to use semantic web technologies, including the Resource Description Framework (RDF) [Ryutov, et al. 2009]. Also, their proposal of a rule-based access control model is based on a constrained first order logic. Although two prior approaches offer models to use ontologies, they fail to provide protection for relations, which is central in the Masoumzadeh and Joshi approach because relations are mainly used in the rule based access control model [Masoumzadeh and Joshi. 2011]. Their approach allows both users and the system to express policies based on access control ontologies. Furthermore, they adopt ontologies to analyze what privacy-sensitive information is protected by the stated policies of the OSN, and find out missing policies for that privacy-sensitive information; moreover, it offers ideal policies to protected users' sensitive information[Masoumzadeh and Joshi. 2013].

In [Fong, et al. 2009, Anwar, et al. 2010, Fong. 2011a], Fong, et al. look to formalize and generalize the access control model implemented in Facebook. The Facebook-style Social Network System (FSNS) is generalized into two steps, namely, reaching the search listing of the resource owner and accessing the resource, respectively. In the first phase, the accessor has to find the owner of the target item; afterward in the second phase, the owner's permissions will decide whether the access is permitted or not. Accessing data and policies for search, communications, and traversal are formalized by this model. One feature of Facebook is that access control policies are topology-based [Anwar, et al. 2010, Fong, et al. 2009]. Although an FSNS is an information-sharing platform that chooses to use an access control mechanism similar to that of Facebook, FSNS' policy vocabulary supports topology-based policies that are not yet offered by Facebook, such as n-common friends and clique. However, the drawbacks of this work are that it lacks support for multiple relationship types, a trust metric of relationships and directed relationships.

In OSNs, it is more natural to take access decisions based on the existence of a particular kind of relationship or particular path of this kind of relationship between the resource owner and accessor. Furthermore, changes in relationships may commonly lead to change in authorizations; thus, access control in OSNs has to tackle the administration of access control policies in addition to normal usage of relationships [Cheng, et al. 2012b, 2012a]. For these reasons, the majority of the access control proposals appearing so far are based on user relationships. The term Relationship-Based Access Control (ReBAC) is a new paradigm for access control that was invented by Gates [2007]. She takes into account the relationship or the transitive closure of the relationship between users and/or resources as essential requirements to protect in Web 2.0 applications. In fact, ReBAC is flexible to support OSN access control requirements, since users can neutrally express their authorizations in terms of relationships. The structures of relationship networks in current OSNs are highly dependent on the type of the social network we are dealing with. In Section 2.1, we described the major online social networks at present but defining the characteristics of the relationship network of each OSN it is not a simple task, because most OSNs merge different types of relationship networks. OSNs, furthermore, may change their focus over time in a way that impacts their relationship network structure. Despite this, we discuss the common and broad characteristics of relationship networks in OSNs. First of all, relationships could be (always) mutual (e.g., the "friend" relationship) in an undirected relationship network. On the other hand, in directed networks, they are not mutual (e.g., the "follow" relationship), where the direction of the relationship denotes which user (node) has established the relationship and the user (node) for whom a relationship has been established. Second, the main feature of a relationship in an OSN is its composition. In particular, binary relations are composable and transitive, for example, considering the relationship friends where we can infer the friends-of-friends relation by the composition *friends o friends*. While a composite binary relationship highly supports policies of composition in relationship-based access control, it can be used to express delegation of trust in a natural way [Blaze, et al. 1996, Weeks. 2001, Li, et al. 2003]. Finally, the type of relationship is often impacted at the level of information disclosure. Thus, authorization decisions may be different in each type; for example, users usually share more private information with close friends and family members than with other types of relationships such as coworkers or colleagues.

Based on the above features that relationships in OSNs have, and inspired by Gates [2007] who, as we mentioned before, takes into account the existence of a relationship or a sequence of relationships between users in social networks to articulate the protection requirements of Web 2.0 applications, several approaches have been proposed to meet those requirements. Following Gates, researchers have offered more advanced relationship-based access control models (ReBAC) and access control policy expressions to talk about relationships. The majority of the access control models that have been proposed to address the problem of access control in OSNs explicitly or implicitly apply the ReBAC paradigm. We begin with the work of Carminati, et al. who proposed the first relationship-based access control model in [Carminati, et al. 2009b], where access control policies are formed as constraints on the relationships between accessor and the item owner. They interpret the constraints in the relationship according to three aspects: relationship type, depth and trust level that are specified by the item owner to grant or prevent access to the resource. Afterward, by using the semantic web technologies including OWL and SWRL in [Carminati, et al. 2009a]the researchers extend the model of [Carminati, et al. 2009b]. The authors used semantic web technology to define user profiles, relationships between users, items that are treated as independent entities, and relationships between users and items, which are extended more than a regular relationship such as ownership. For example, the relationship between a user and a photo that she/he is tagged in is expressed as ‘photoOf’ in their language. In [2009]Fong, et al. developed a formal algebraic access control scheme to model Facebook privacy settings. They model the access control procedure as two stages: reaching the search listing of the item owner and accessing the item. Their access control policies are primarily based on the relationships between the accessor and the item owner and graph-theoretic properties such as “n-common friends” and “n-clique”. However, this model does not support direct relationships, various relationship types (only friends) such as user-to-resourse or resourse-to-resourse relationship and trust metric of relationships.

Afterward, Fong [2011b]formulated a model for social computing applications, in which authorization decisions are based on the relationships between the resource owner and the resource accessor in a social network to meet Gates' requirements of protection in Web 2.0 applications [Gates. 2007]. Moreover, Fong uses a modal logic language for policy specification and composition and tracks social networks that are poly-relational. Later, Fong and Siahaan examined and demonstrated that there are well-known relational policies that cannot be articulated in the ReBAC policy language. Consequently, they extended and improved the model language to facilitate the specification and composition of ReBAC policies [Fong and Siahaan. 2011]. In opposition to earlier work [Fong, et al. 2009], these two works support multiple relationship types and directional relationships; also, the relationships and authorizations in these models are formalized in access contexts and a context hierarchy [Fong and Siahaan. 2011, Fong. 2011b]. Last but not least, these models were enhanced by Bruns, et al.[2012]who adopted a hybrid logic to further facilitate and make efficient policy evaluation and specification. From the perspective of helping online users to analyze their access control policies, Anwar and Fong designed a visualization tool to evaluate the consequence of the access control formation [Anwar and Fong. 2012].

User-to-user relationship-based access control (UURAC) is a model proposed by Cheng, et al. [2012b]who also provide an expression-based policy specification language, which offers further advanced and fine-grained access control in OSNs. However, the UURAC model supports only one relationship type for authorization and specification of policies, namely the user-to-user relationships type. To improve UURAC limitations, Cheng, et al.[2012a]later proposed a rich social network model that includes user-to-user, user-to- resource and resource-to-resource relationships. This work considers the resources as entities, which is similar to the one in [Carminati, et al. 2009a], and the actions that users perform on their resources are recognized as relationships. Moreover, users' administrative activities beside normal usage activities are supported. By explicit treatment of user-to-user, user-to-resource and resource-to-resource relationship-based policies, access control policies and conflict resolution, they significantly extend the previous UURAC model. Applying conflict resolution policies over relationship precedence is the strategy used to address policy conflicts in [Cheng, et al. 2012a].

Recently, a new system has appeared to enable online users to think differently about their privacy in OSNs [Pang and Zhang. 2013]. In contrast to [Cheng, et al. 2012b, 2012a], Pang and Zhang proposed a new access control scheme for OSNs that focuses on existing public information in OSNs. They proposed a new manner for users in OSNs to express their privacy requirements based on their connections through public information. By comprising both a user relationships network and public information network they provided a new social network model for online users to regulate access to their resources. Afterward, authors developed a hybrid logic to be used for expressing access control policies. They offered a number of policies based on relationships and public information and articulated them in their developed logic.

2.3.2.2.2 Access Control Models with Multiple Specifications (Co-owners)

In response to the need for joint management for data sharing in OSNs, Squicciarini, et al. [2009, 2010]have provided a novel collective privacy mechanism for better managing shared content between users. Their work considers the privacy control of content that is co-owned by multiple users in an OSN, such that each associated controller may separately specify her/his own privacy requirements for the shared content. They categorized associated controllers into two groups: first, owners are those users that manage access to the shared content and they can be a single user or a group of users. Moreover, they own and share ownership authorities with the originator who originally creates or uploads a particular content to the OSNs. Also, this work considers users who request access to certain content as viewers. The Clarke-Tax mechanism (a voting algorithm) is adopted to enable collective enforcement for shared data. Furthermore, a game theoretical method [Grossklags, et al. 2008], based on the voting algorithm, is applied to evaluate the scheme and to consider the privacy requirements of all stakeholders. Building upon the Clarke-Tax mechanism, the time needed to implement the algorithm is so little, that the collaborative management of the privacy settings is transparent to the user. In their work, by determining the maximum depth of viewers in the social network graph, each co-owner can identify privacy policies for her/his contents. For example, if an owner regulates her/his access requirement with a maximum depth=1, the allowed viewers are equivalent to direct friends, while a maximum depth of 2 would equate to friends of friends. Although their contributions include a solution for policy conflicts among multiple owners and inference techniques that free the users from the burden of manually regulating privacy preferences for each content, automatic negotiations adopted in their approach offer limited capacity for negotiation. Moreover, one of the possible side effects could be caused by the auction process that only the winning attempts could control who can access the shared content, instead of harmonizing all stakeholders' privacy preferences. For example, owners cannot protect their content if other co-owners strongly request to publish this content. Their method is not as simple as it is claimed to be; it could be actually hard for ordinary OSN users to understand the Clarke-Tax mechanism and specify appropriate bid values for auctions which are essential for their process to derive a collaborative decision. Lately, CoPE, presented by Squicciarini, et al. [2011a] as a system, provides users with privacy mechanisms to collaboratively control and protect the access to their data. This application is implemented within the context of Facebook to support the collaborative privacy-control mechanisms that offers the ability to all stakeholders to manage accessibility of shared data.

Carminati and Ferrari in [2011]discuss access control for data sharing in social networks, with emphasis on conflict resolution in circumstances where multiple controllers are involved. In particular, they introduce a collaborative access control mechanism in OSNs that integrates the topology of social networks in policy-making. They improve topology-based access control taking into account a set of collaborative users by giving a new class of security policies, called collaborative security policies, which indicate the set of users who should contribute to the collaboration. Additionally, Carminati, et al.[2011]propose a model that employs semantic web technologies to support a rich way of symbolizing collaborative users.

Similarly, with respect to collaborative access control, the authors of [Wishart, et al. 2010]offer policy-based approaches to control access to shared data in social networking applications. Their collaborative approach to authoring privacy policies takes into account the needs of all associated users who are affected by disclosure of content. The originator of the content on the social network is allowed to specify policies for the content she/he uploads. Then, the policy application can be edited by users who are nominated from the social networking service. In their approach, all requests that came from users, who are interested in the access or dissemination of the content, are passed to the Policy Decision Point (PDP). This PDP evaluates the policy for the content using a knowledge base. Although Wishart, et al.[2010]work is one of the few solutions to offer user-interfaces, a general drawback of their solution is the limited ability to determine co-owners. The authors choose to place the burden of specifying policies for the content and inviting co-owners solely on the uploader of the content. They do not offer capacity for users to claim co-ownership of content or mechanisms to realize which user has the right to participate as co-owner.

There are also some works discussing the collaborative privacy management problems in the popular social networks. CAPE, presented by Xiao and Tan [2012], takes into account peer effects in making collaborative access control polices; thus they believe some conflicts of co-managers' intentions will vanish naturally. Their proposed framework, CAPE, is based on graph theory and their ideas behind integrating peer effects is that OSN users are connected and immensely influenced by their neighbors. Consequently, by simulating such social interaction automatically, they allow co-owners to adjust their privacy settings according to their neighbors' actions and intentions. However, a general drawback of their approach is formulating their model based on an emotional mediation among multiple co-owners. It is difficult to trust and validate such a claim that co-owners will change their privacy settings according to their neighbors' intentions. Moreover, they do not offer a solution to deal with the conflict that may be caused by a malicious co-owner who could subversively affect other co-owners' actions.

Sun, et al. [2012] proposed a recent approach that discusses the design of access control mechanisms for protecting shared data where multiple co-owners may have differing and conflicting privacy requirements. They offer a trust-augmented voting scheme to solve the particular problem of how to merge diverse privacy requirements from co-owners of shared contents. The core idea in their approach is combining trust relations among users in OSNs and Condorcet's voting schemes [Young. 1988], where trust values are considered as vote weights. They believe that trust should be a key contributing factor to be considered when multiple co-owners collaborate to decide the privacy policy on a shared content. In addition, they choose voting as a natural way to construct a mechanism that takes individual's privacy preferences into a joint decision reflecting the collaborative privacy intentions of the group of owners who are sharing ownership of particular content. Although their approach is based on the fact that trust is naturally inherent in OSNs and that a preferential voting scheme is a meaningful and straightforward way for co-owners to formulate their privacy requirements on shared data, the owner solely can decide for her/himself how she/he wants the privacy setting of shared content to be when a co-owner's decision is overridden. Also, if an owner is not satisfied with the decision produced by a vote, she/he can solely specify the privacy policy for shared content. So under these circumstances the drawback of their solution is that the decision for regulating the access to the shared data still rests only on one owner who is the uploader of shared content or the owner of the space where the shared content is.

Still in the field of collaborative access control in OSNs, Hu and colleagues proposed several works to address the privacy risks that are caused by the limited access control mechanisms in current OSNs, which do not provide effective mechanisms to enforce privacy concerns over data associated with multiple co-owners. First, in [2011]Hu and Ahn proposed a multiparty authorization framework (MAF) that enables collaborative management of shared content in OSNs. MAF is formulated to capture and realize the core features of multiparty authorization requirements in OSNs. They explored the privacy risk of lacking collaborative access control for data sharing in OSNs, which could sap the users privacy. Moreover, they combined MAF with a multiparty policy specification scheme and corresponding policy evaluation mechanism. MController is a proof-of-concept implementation of their approach that is deployed as a third-party application on Facebook, along with performance analysis. In fact, it is not uncommon that conflicts will arise when we attempt to reach a collaborative decision. To meet this requirement, Hu, et al. have studied data sharing in social networks, with emphasis on conflict resolution in the case of multiple co-owners involved in regulating shared data policies [Hu, et al. 2011]. They offer a novel solution for detecting and resolving privacy conflicts for collaborative data sharing in OSNs. Their systematic conflict detection and resolution mechanism balances the need for privacy protection and the online users' desire for data sharing by quantitative analysis of privacy loss and sharing risk. Privacy conflict identification can be realized through specifying the privacy settings to reflect the privacy requirements; furthermore, in a privacy conflict identification, the authors adopt an algorithm for identification of conflict. Also, in their paper, they discuss several situations of privacy conflicts for understanding the risks caused by those conflicts. Finally, to implement the approach they have designed a third- party Facebook application called Retinue which is in charge of the privacy conflict detection and resolution, and the production of a conflict-resolved privacy policy, which is then used to decide who has the access rights to the shared data. However, the above-mentioned approaches to address privacy policy conflicts and collaborative management of shared data still need improvements. We believe that before the decisions are created, we have to set a negotiation mechanism for conflicting privacy policies. By this suggestion, users may become aware of what data about them will be exposed; this knowledge can lead them to address some privacy policy conflicts.

In addition, Hu, et al. [2013]significantly improve the multiparty authorization framework by presenting data sharing patterns with respect to co-controllers' authorizations in OSNs. Also, they have enhanced a method to represent and reason about their model in a logic program. By these improvements they have formulated the Multiparty Access Control (MPAC) model to grasp the main features of multiparty authorization requirements that have not been actually accommodated so far by any existing access control systems and models for OSNs. Furthermore, they have enhanced a policy specification scheme and a voting-based conflict resolution mechanism. Their proposed conflict resolution mechanism assembles each co-owner's decision policy and sensitivity towards a particular content and hence leverages each co-owner's preference in the process of making a collective decision. Moreover, they present methods to perform analysis on the access control model such as correctness analysis and authorization analysis. The research papers [Hu and Ahn. 2011, Hu, et al. 2013, 2011] support some theories that we are extended in this dissertation. However, the appropriate strategy for conflict resolution, in their solution, is selected by the single owner who has the shared content in her/his profile [Hu, et al. 2013]. Thus, they do not treat all associated controllers equally in the process of making a collective decision. Even if the owner desires to have highest priority in the control of shared data, we believe it may be more appropriate if all co-owners participate in the process of selecting the strategy for conflict resolution. Although they have defined types for the related co-owners based on their relationships with the shared content, those controllers have the same influence in the process of making a final decision. However, we believe to have collaborative decision-making be both efficient and effective, we have to give a weight to each ownership type that represents the priority of this type of ownership in the control of shared data. As a result, their privacy conflict resolution approach needs more comprehensive investigation. In addition, their analysis of data sharing associated with multiple users in OSNs is limited, and they did not inclusively articulate all scenarios of privacy conflicts for understanding the risks posed by those conflicts. We believe that, in [Hu, et al. 2013], they need to explore more criteria to evaluate the features of their solution for collaborative management of shared data.

With respect to a multiparty access control (MPAC) model, Hu, Ahn, and Jorgensen propose and implement an approach for collaborative management of shared data in Google+ [Hu, et al. 2012]. This is despite the fact that Google+ has the notion of circles that allow users to selectively share data with certain groups within their social network, instead of sharing with all the users in their social networks [Kairam, et al. 2012]. However, Google+ still offers limited access control that only supports a single owner to regulate the access policy of the shared content. Consequently, they expand and articulate a collaborative access control model called MPAC+, to determine the essentiality of collaborative authorization requirements in Google+, along with a policy specification scheme and conflict resolution mechanism, which interacts with conflicts of privacy requirements by keeping the balance between the need for privacy protection and the users' request for data sharing, for collaborative management of shared data in Google+. They end by giving a prototype implementation of their collaborative privacy management approaches, called Sigma. Since the MPAC+ approach is built on the similar concept of MPAC, in general they have similar drawbacks. Furthermore, their approach has to be implemented and evaluated in a Google+ platform to show more precise and accurate results.

Photo sharing (tagging) is the most important service in many online social network sites that needs joint management. Thus, it has been recognized by recent works [Besmer and Richter Lipford. 2010, Pesce, et al. 2012, Squicciarini, et al. 2009]. In [Besmer and Richter Lipford. 2010], the authors believe understanding users' privacy concerns and desires leads them to design an efficient collaborative access control mechanism. So they have investigated users' privacy concerns about a photo tagging service and proposed a set of design considerations for a tagged photo privacy management approach. Then, they have created a privacy mechanism attempting to address those needs and follow those considerations. One of their findings is the issue of photo ownership is very significant and relevant to the photo tagging service; thus they provide a mechanism as part of a collaborative management tool in a way that benefits all tagged users. Although the authors report a comprehensive understanding of privacy concerns and requirements of users and found a number of important design considerations for a photo tagging privacy mechanism, their investigation and findings are only based on Facebook. Consequently, the concerns and findings of users' privacy requirements they discovered may not be applicable to other social network sites such as MySpace and Google+.

2.4 Trust in Online Social Networks

As introduced in section 2.1, OSNs nowadays have become essential activates in our daily life. OSNs reflect human relationships on the Web, allowing users to establish new relationships and maintain existing relationships. Then, they connect to other users they know, to share information, interests and to have group activities (e.g., games, events). Since the literature demonstrates that social life is simply impossible without trust [Schlenker, et al. 1973, Luhmann. 1979], trust becomes one of the most crucial concepts in online social networks and online communities for improving privacy mechanisms and reducing concerns about personal information disclosure.

Trust has been defined in several different disciplines such as psychology, economics, sociology and computer science. The concept of trust in these different disciplines differs in how it is represented, computed, and used. Thus, there is a wide and diverse range of synonyms for trust, and the answer to "what is trust?" cannot be easily offered. Generally, the verb trust can mean to have confidence or believe in something or someone such as the honesty, skill or safety of a person, organization or thing [Cambridge Dictionaries online. 2014]. In computer science, most of the works regarding trust have been concentrated in the area of security, and then it has been rapidly applied to other areas such as game theory and electronic commerce. Consequently, numerous definitions have been offered seeing trust from diverse viewpoints; however, these explanations may not be directly related to OSNs. Although the majority of the studies concerning trust in OSNs have used mathematical techniques to develop and verify OSN services, the meaning and modeling trust in OSN are significant challenges.

To simulate trust in OSNs, we have to keep in mind three main properties of trust. Based on [Golbeck. 2006, Golbeck and Hendler. 2006, Sherchan, et al. 2013, Golbeck. 2005], the main characteristics of trust can be defined as follows:

- Asymmetric: Between two users, the trust level is not identical; that means trust is not necessarily the same in both directions. For example, Alice may trust Bob 80%; however, Bob may not have the same amount of trust feeling about Alice. Bob may only trust Alice 20% in return for example. In the undirected social graph, which is the basis of friendship oriented networks (e.g., friends in Facebook), this property is difficult to capture.
- Transitivity: Trust is not perfectly transitive in the mathematical sense where if Alice highly trusts Bob, and Bob highly trusts Carol, then it is not necessarily true that Alice so highly trusts Carol. There is a notion that trust can be passed between people [Golbeck. 2005]. Let us suppose that Alice and Bob know each other very well. When Alice asks Bob for an opinion about a hotel, Alice considers Bob's opinion then integrates that to help shape an initial opinion of the hotel. Likewise, an OSN user can distinguish the most appropriate content depending mostly on her/his friends' past experiences.
- Personalization: Trust is inherently a personal opinion. In fact, two people often have very different opinions about the trustworthiness of the same person. For example, Alice and Bob may have very different opinion about Carol, but actually there is no right or wrong trust value except from the perspective of Alice and

Bob. According to Deutsch's definition: we commit to take the ambiguous path if we believe that the trusted person will take the action that will produce the good outcome [Deutsch. 1962]. Whatever qualifies as a suitable outcome varies from one person to another.

On existing OSN graphs, a label is assigned to each link to symbolize the trust value of the relationship. In other words, users can explicitly assign a trust value to those who they have direct relationships with. Subsequently, a user can often make a decision based on this trust value of others and their opinions. On the other hand, when users are not directly connected, a seeker needs know the trustworthiness of data and ensure it is not from a malicious user who may give false information that might lead to disclosing private information. An important problem of trust in OSNs is to determine how much one user in the trust graph should trust another one who is not directly connected to her/him. The issue of trust is treated by an approach called trust inference. This problem of trust is illustrated in Figure 2. The node that denotes the individual who requests to compute her/his trust value to another one is called the source. In contrast, the node that the source requests to deduce about is called the sink.

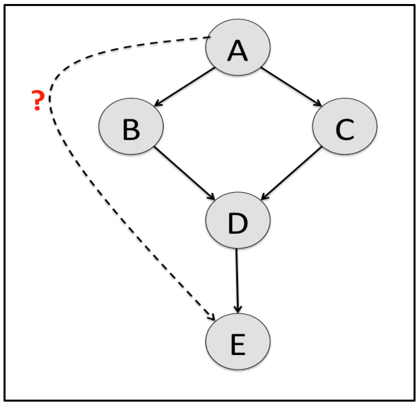


Figure 2: Trust Inference from node A to node E.

As shown in Figure 2, we can say A, or the “source”, is directly linked with B and C, but is not directed linked to D or E. Moreover, we can recognize A is indirectly connected to E through two paths, A->B->D->E and A->C->D-> E, therefore generating two trust values of the “sink”, E, when we are deciding the trust inference value of A to E.

In the context of OSNs, a trust inference mechanism can be described as the ability to determine how much a user (source) trusts the sink when the user does not know the sink directly. It can be used for aggregating, filtering, and ordering of information [Sherchan, et al. 2013]. There are a number of techniques that are proposed by scholars around the world to find suitable algorithms for inferring the optimum path and the trust value. The most common algorithms for trust inference include TidalTrust [Golbeck. 2005], SocialTrust [Caverlee, et al. 2008], SUNNY [Kuter and Golbeck. 2007] and FuzzyTrust [Lesani and Bagheri. 2006, Lesani and Montazeri. 2009]. We believe the notion of trust is naturally present in OSNs, and furthermore, users naturally tend to use linguistic expressions when they are asked about their trust to other individuals. For these reasons, we adopt the FuzzyTrust algorithm in our approach for calculating trust values, as coined by [Lesani and Bagheri. 2006]. In the following, we discuss the FuzzyTrust algorithm and TidalTrust algorithm because it is used as the basis for FuzzyTrust algorithm.

In Golbeck's work [2005], additional to deducing some trust graph properties from real networks, the author proposes the TidalTrust algorithm to derive a trust value between two users in the social network using the FOAF vocabulary. In this algorithm the trust values are considered to be numbers in a continuous range of [0…10], where each neighbor of the source is assigned a particular trust value. Afterward, the source node searches the path from her/his node to the sink by votes from all its neighbors, then paths are estimated and the shortest path is used. Golbeck assumes that neighbors with higher trust ratings have mostly concurred with each other in the trustworthiness of the information source (sink). Consequently, shorter paths have a lower average difference and higher trust ratings have a lower average difference. As a result, the approach takes into account the shortest paths from source to sink as the most accurate path. The TidalTrust algorithm is simple and its low complexity $(O\ (|V|\ +\ |E|))$ is suitable to any social network requiring high scalability such as online social networks.

The FuzzyTrust algorithm is proposed by Lesani and Bagheri [2006] to tackle the problem where trust inference in a large trust graph is faced with contradictory information. They offer fuzzy linguistic terms to assign trust ratings to other nodes in the trust graph and developed an algorithm based on these. The fuzzy membership functions for the linguistic terms such as low, medium, medium low, medium high and high can be used, which makes it easier for users to specify a trust value. Similar to the TidalTrust algorithm, this algorithm adopts the shortest path assumptions for trust calculation. Moreover, it computes trust from a stronger path when we obtain more than one path having the same depth from source to sink because Golbeck [2005] shows that paths with higher trust ratings produce better trust inference. The FuzzyTrust algorithm [Lesani and Bagheri. 2006, Lesani and Montazeri. 2009] performs a breadth-first like search through the nodes to detect the shortest and strongest path. Although this algorithm uses the TidalTrust algorithm as a basis, the results of Lesani's and Bagheri's simulation show that the FuzzyTrust algorithm comes up with more accurate information than TidalTrust.

Chapter 3

3 Collaborative Access Control Scenarios

In this chapter, we explain scenarios where more than one user should be involved in the access control process. Multiple controller scenarios are raised by using different tools in OSNs such as posting, tagging and sharing. Often, multiple users have a variety of privacy policies over shared content; however, existing access control mechanisms in OSNs choose to place the burden of privacy policy solely on the owner who has the shared data in her/his profile. To safeguard all associated users' privacy in OSNs, collaborative privacy protection mechanisms are needed. First we introduce and analyze multiple controllers' scenarios to determine all users who have the right to participate in the process of making a collective access control policy. Those analyses and determinations are critical to the success of making collective privacy management of users' shared content. To clarify the scenario analysis, we categorize them into three types: profile sharing where accessors are the social applications, relationship sharing where a relationship between users represents a shared item that users may have different authorizations concerning who can know about it, and content sharing which is the main type of sharing pattern and has the most sub-categories. Thus, we precisely analyze all cases and subcases of content sharing patterns. We employ Facebook as the running example to make our explanation of sharing patterns easier. Although we use Facebook for examples, our discussion could work for other online social networks, such as Google+.

3.1 Profile Sharing

Several OSNs are able to provide open platforms to enable any third-party developers worldwide to create full applications on the top of users' profiles, inside the framework. To enable social applications to be more purposeful and meaningful, they may consume user profile attributes, which usually include information such as the user's name, birthdate, status, address, emails, education, interests, photos, music, videos, and many other attributes. At the same time, third-party applications could be extended and use attributes of user’s friends, which would pose serious privacy concerns for the friends. When users and their friends use the same application, both the user and her/his friend want to control which attributes the application can access. Current OSNs allow only one side of the relationship to govern access to the profile attributes of the other side. Hence, the decision of an accessing application is solely regulated by the user who desires to share her/his friend information with a third party application in the OSN. To address such a critical issue, we consider the user’s friend is an owner who owns shared data on her/his space, which consists of profile attributes. The second controller is a contributor who shares her/his friend’s profile attributes with a social application. Then, we offer a mechanism to combine owner and contributor privacy settings of the shared profile attributes. Figure 3.displays a profile sharing pattern. Our proposed solution, to be introduced in Chapter 4, PermittedandDeniedAccessors, with its necessary inputs, is shown intervening between the application and the data to be controlled.

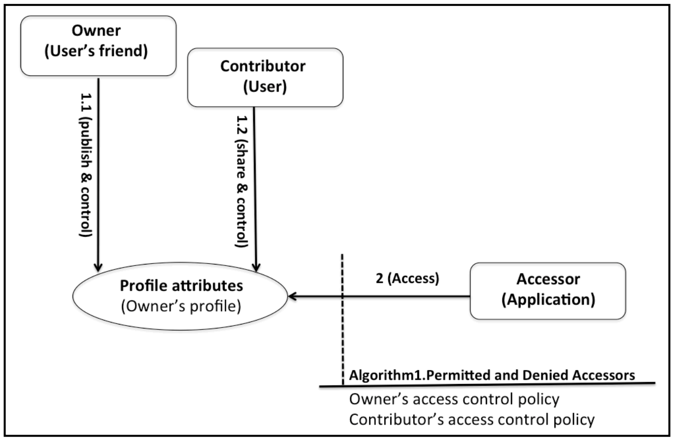


Figure 3: Pattern of profile sharing.

3.2 Relationship Sharing

Users in OSNs are connected by social relationships, which characteristically are bidirectional. OSNs enable users to share their relationships with other members in OSNs. In fact, there are two users who establish the relationship in OSNs; consequently, both of them have the right to manage who can see the relationship between them. However, current OSNs provide limited access control that allows only one side of a relationship to restrict access where the user on the other side of the relationship may have a different privacy preference. Consequently, the result of current OSN access control causes a high level of disclosure in online relationships because the participants in a relationship may have dissimilar sensitivity levels with respect to each other. The associated users in this case are co-owners. The need for a solution addressing the problem of relationship information leakage is demonstrated in Figure 4, where the relationship is the shared item between two users. The first user is called a stakeholder who specifies a policy to hide her/his relationships from the public. The second user is an owner who adopts a weaker policy that allows the public to see her/his relationships list. In this scenario, to regulate a satisfactory policy we have to consider owner and stakeholder authorization requirements to achieve a collective decision. Again, our proposal for handling this, PermittedandDenied Accessors, which is given in Chapter 4, is shown intervening between the accessor and the data.

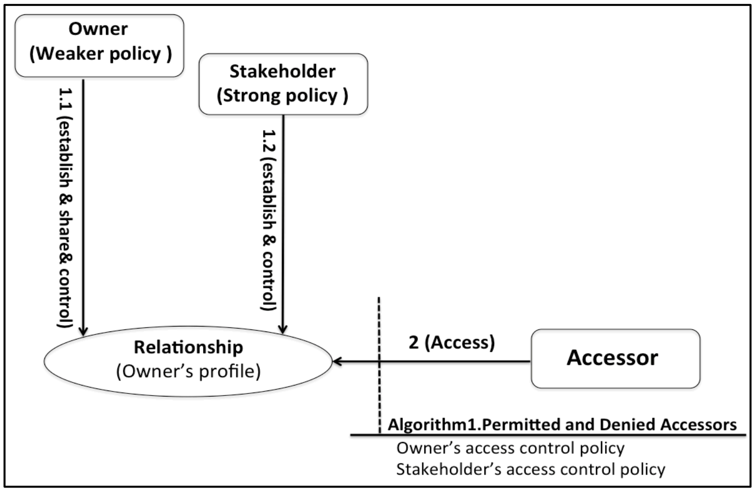

Owner
(Weaker policy)
Stakeholder
(Strong policy)
1.1 (establish & share& control)
1.2 (establish & control)
Relationship
(Owner's profile)
2 (Access)
Accessor
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Stakeholder's access control policy

Figure 4: Pattern of relationship sharing.

3.3 Content Sharing

OSNs offer mechanisms that facilitate users to socialize in the digital world. The main purpose of relationships in OSNs is to share various resources, which includes information, photographs, music and videos. Posting, tagging and information exchange are sharing tools that are provided by many OSNs such as Facebook, Google+ and My Space. In this section, we introduce content sharing scenarios for which privacy policies in current OSNs do not adequately provide collective privacy controls on shared content. First, a user is able to post notes and news in her/his own space, upload pictures and videos, tag others members in her/his contents and share her/his contents such as pictures, videos, news etc., with other users. Furthermore, OSNs allow users to post contents on their friends' profiles and share their friends' contents. We organize the scenarios for content sharing depending on the sharing tools that are applied to the content.

3.3.1 Tagging

The tagging is the most popular social networking features. Tagging a user not only facilitates users to organize their photos, but also urges users to share and disseminate photos in OSNs. However, tagging carries several questions about what objects the tagging refers to, and who are the interpreters. Presently, OSNs give the user being tagged permission to accept the tagging or remove it; however, the photo is still in the OSN. Thus, current access controls in OSNs offer limited support to tagged users, who may be explicitly identified through tags, because the holder of the tagged photo is the sole decision maker to regulate access over who can see and share the photo. To limit disclosure of information, we choose to divide the burden of privacy setting among all associated users who appear in the photo and the owner of the profile where photo is shared. Thus, we can reach a collaborative decision, which is represented by the result of our PermittedandDeniedAccessors algorithm that considers privacy settings of all co-owners. By making a cooperative decision, each associated controller can declare her/his desire about who can access to the photo and who cannot.

Let us explain the tagging scenario by example where a picture contains four users, Alice (A), Bob (B), Carol (C), and David (D). First, Alice uploads the photo to her profile and tags Bob, Carol, and David in the photo. We consider Alice as the owner of the photo, and Bob, Carol and David as stakeholders. In this case, the owner, who launches the action using the tagging tool, and the stakeholders have to specify their access control polices to restrict who can view the photo. Figure 5 below illustrates the tagging scenario.

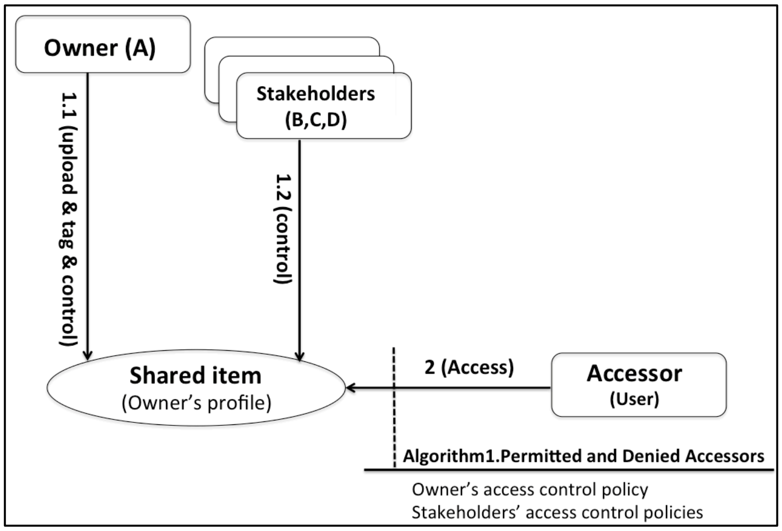

Owner (A)
Stakeholders (B,C,D)
1.1 (upload & tag & control)
1.2 (control)
Shared item (Owner's profile)
2 (Access)
Accessor (User)
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Stakeholders' access control policies

Figure 5: Tagging scenario.

3.3.2 Posting

To encouraging users to share data with others, OSNs provide another sharing tool, posting, where user can post content in someone else's profile. When a user posts an item in someone else's space, she/he becomes the contributor. The owner of the profile where the contributor posted the content is called the owner. In current OSNs, access control supports privacy decisions as an individual process. Then in a posting situation, the decision maker is only the owner who receives the posted item in her/his profile. Consequently, the contributor, who is the original uploader, is not able to regulate who can access her/his posted content, which may lead to violations of her/his privacy. To address this limitation, we analyze the scenario where all associated controllers are able to participate in the access control decision. By combining all controllers' privacy requirements, we offer collaborative access control rules that are represented by the result of our PermittedandDeniedAccessors algorithm.

Let us consider a more complex example, where Alice not only posts a photo in Bob's profile but also tags Carol in this photo. We call Alice, who uploads the photo, a contributor. Bob, whose profile is the location of the shared photo, we call an owner and Carol is considered to be a stakeholder. Alice (A), Bob (B), and Carol(C) should cooperatively manage the access to their shared photo. Figure 6 shows this posting scenario where the contributor posts content in another user's profile and the content may have multiple stakeholders. PermittedandDeniedAccessors is again intervening, with appropriate inputs.

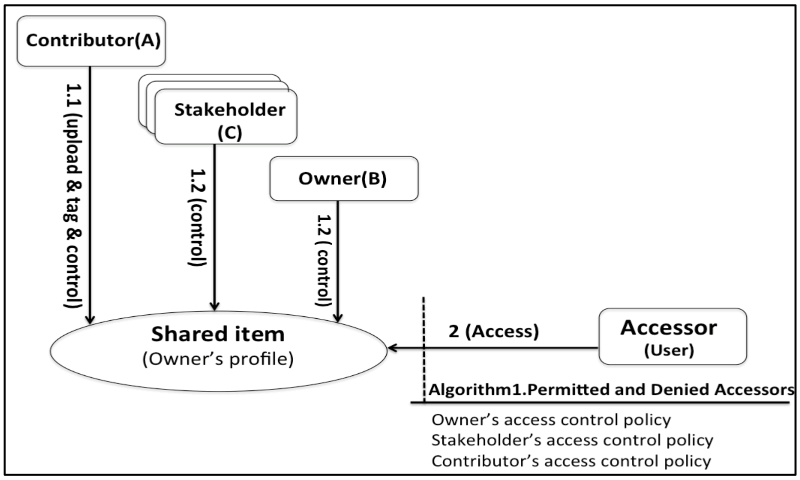

Contributor(A)
1.1 (upload & tag & control)
Stakeholder (C)
1.2 (control)
Owner(B)
1.2 (control)
Shared item
(Owner's profile)
2 (Access)
Accessor
(User)
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Stakeholder's access control policy
Contributor's access control policy

Figure 6: Posting -Tagging scenario.

3.3.3 Sharing

The sharing tool supports distributing data among members in OSNs in various ways. Users can share their contents with others in their social network; otherwise, users can share others users' contents. Also, users can share other user's content and post it in someone else's space. In general, whenever types of sharing apply to an item in OSNs there is high potential of identification all linked users who are related to this item. Current OSNs, until now, provide individual processes to make a decision over who can access the shared items. As a result, those items, which obviously expose the identity of all associated users, may violate a users' privacy and lead to their embarrassment. We believe it could be more practical and reliable to allow all linked users to participate in the privacy setting of a shared item. To reach this goal, first we introduce and analyze three multiple controllers' scenarios that are raised by using the sharing tool, then we can solve the problem of how to merge privacy opinions from co-controllers of shared items.

3.3.3.1 Sharing user's content with other users

For the first multiparty sharing scenario, when a user shares her/his content with others, the content will be in turn be posted in another's profile. Presently in OSNs, the regulator deciding who can access the shared item is only the owner of the profile. Consequently, the original user who shared the content will lose control over it. For instance, when Alice (A) desires to share her photo with her friend Bob (B), the photo will be in turn posted in Bob's profile. Intuitively, the decision over who can access this photo should be regulated by Alice and Bob. In this situation, we call Alice the originator and Bob, who has the shared photo in his space, the owner. The initial access control policy of this item reflects the originator's privacy requirements; furthermore, the privacy setting of the new owner should impact the final access control policy of the shared content. By our PermittedandDeniedAccessors algorithm, we merge the owner's and originator's privacy requirements to achieve final and collaborative access control polices of shared content. Figure 7 demonstrates the content sharing pattern that is generated by using the sharing tool where users share their contents with others members.

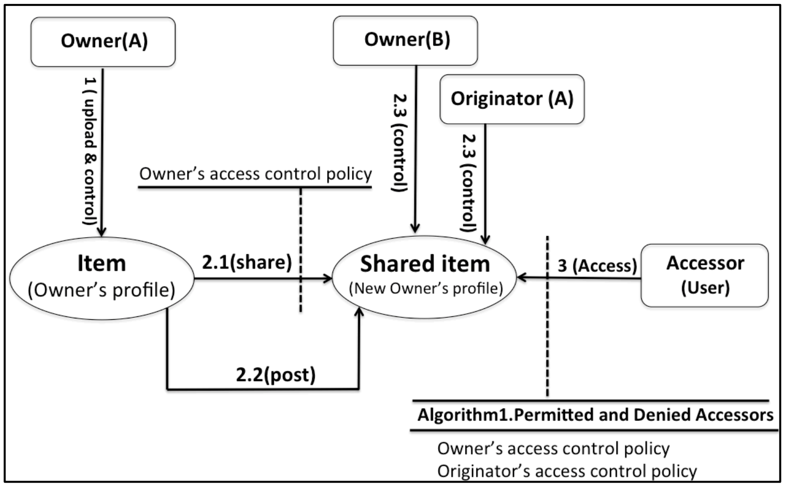


Figure 7: Simple sharing users' contents with other user scenario.

Under the case of sharing users' contents with others scenario there are two subcases. The first is when an item that is desirable to share, is linked with stakeholders through tags. Therefore, the sharing request could become from the owner or one of the stakeholders according to our analyses of the tagging scenario. OSNs allow tagged users to share contents that have tags in them to other users. In current OSNs, the tagging user is not required to ask for permission of the other tagged users or the uploader, when she/he desires to share content with her/his social network. Also, when the user who published the item and tagged other users in it wishes to share content with someone else, she/he is not required to ask tagged users, who are appearing in the item, permission. In general, most OSNs offer access control that supports only a single user's privacy requirements. For this reason, others associated users do not have the ability to control their data and they cannot influence the privacy policy applied to this data. In response to this issue, we are going to analyze the sharing-tagging scenario to capture users who should be involved in the process of making a collective decision about disseminating a shared item. Our Controller Sharing algorithm is a method to achieve the cooperative decision. To clarify this further, let us suppose the scenario where Alice (A) wishes to share a photo with her friend Bob (B), and Carol (C) and David (D) were tagged in this photo. Alice's desire may cause exposure of Carol's and David's privacy; consequently, collaborative access control may be more intuitive. In our example, we classify Alice as the owner of the photo and Carol and David as stakeholders; in fact, all of them should have some impact on the final decision of Alice's sharing request, even Alice. Alternatively, a user who is tagged can further share the photo with her/his social network. Consequently, a sharing request could come from the owner or stakeholders. As a result of the tagging-sharing users' contents with others scenario, the photo will be in turn posted in Bob's profile and he can specify an access control policy as owner. Also, the original user who shared the photo with Bob, in our example Alice, should control who can access the photo in Bob's profile as originator. Figure 8 shows the tagging-sharing users' contents with others' use case that is generated by sharing an item that is originally tagged scenario.

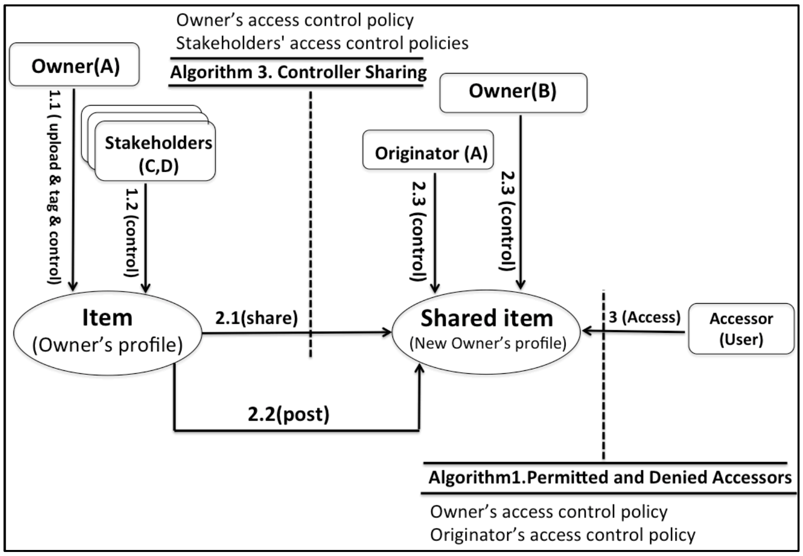

Owner's access control policy
Stakeholders' access control policies
Algorithm 3. Controller Sharing
Owner(A)
1.1 (upload & tag & control)
Stakeholders (C,D)
1.2 (control)
Owner(B)
Originator (A)
2.3 (control)
2.3 (control)
Item
(Owner's profile)
2.1(share)
Shared item
(New Owner's profile)
3 (Access)
Accessor (User)
2.2(post)
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Originator's access control policy

Figure 8: Tagging-sharing users' contents with other user scenario.

The second subcase of sharing users' contents with others scenario, which is more complicated, occurs when users who may be the owner or stakeholders wish to share their contents with others, and that content originally was posted by someone else in the owner's space. As we mentioned previously, current OSNs offer limited access control support - only a single decision maker may restrict access over shared contents, regardless of other users' privacy concerns, even if they essentially are linked or appear in the contents being considered. For example, we will analyze the posting-tagging-sharing users' contents with other users scenario. Also, we will determine users who should have the right to be involved in the process of making a collective decision about who can disseminate the shared item.

Consider a scenario where, Alice (A) wishes to share a photo that originally was posted in her profile by Dave (D) who tagged Carol(C) in the photo. Intuitively, each of those users would want to participate in the process of making a cooperative decision about Alice's request. In response to this desire, we have classified associated users as follow: Alice is the owner, Dave is a contributor and Carol is a stakeholder. When Alice's sharing request is permitted, the photo will be in turn posted in Bob's profile. Hence, the reposted photo is controlled by Alice as originator and Bob as owner. Note that Dave and Carol are not involved in the process of making a decision over who can access the photo in Bob's profile because they already offered permissions to Alice to share the photo with her friend Bob. Below, Figure 9 demonstrates a posting-tagging-sharing users' contents with other user scenario that is generated by sharing content that was originally posted and has tagged users.

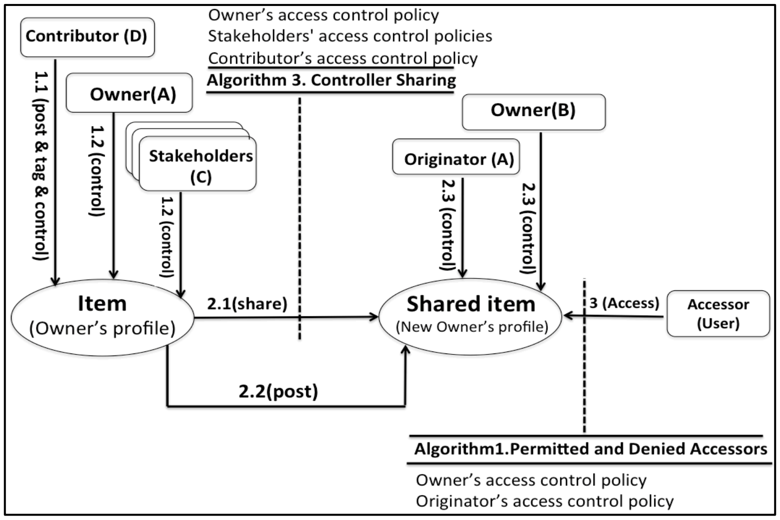

Owner's access control policy
Stakeholders' access control policies
Contributor's access control policy
Algorithm 3. Controller Sharing
Contributor (D)
Owner(A)
Stakeholders (C)
Owner(B)
Originator (A)
1.1 (post & tag & control)
1.2 (control)
1.2 (control)
2.3 (control)
2.3 (control)
Item
(Owner's profile)
2.1(share)
Shared item
(New Owner's profile)
3 (Access)
Accessor
(User)
2.2(post)
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Originator's access control policy

Figure 9: Posting-tagging-sharing users' contents with other user scenario.

3.3.3.2 Sharing others users' content

OSNs encourage users to share more data by enabling them to share others' contents. While, OSNs' users can share their contents with others users, they can request other users to share their contents. When a user shares others users' content, it will be in turn posted in her/his profile. In current OSNs, a reposted item is only controlled by the owner of the profile; thus, the original uploader will lose control over who can access the reposted item in the new owner's profile. We believe that users who are connected with an item should be involved in the process of making shared item access control policy. For this reason, we are going to analyze all cases of sharing the others users' contents scenario.

For the simple case, let us assume that a user, say Alice (A), views a post in Bob's profile and desires to share it with her social network. If Bob (B) allows her to share his post with her social network, the post will be in turn posted in her profile. Then, Alice, who disseminates the post, can regulate her privacy policy as owner for this post, and the original policy that was adopted by Bob, who is the originator now, should influence the final access control policy of the reposted content. By classifying all users connected to a shared item, we can consider the privacy settings of all of them to determine the collective decision on the access restrictions over shared content. We will discuss how our algorithm PermittedandDeniedAccessor works to reach acollective decision in the next Chapter. The simple case of sharing others' contents scenario is shown in Figure 10.

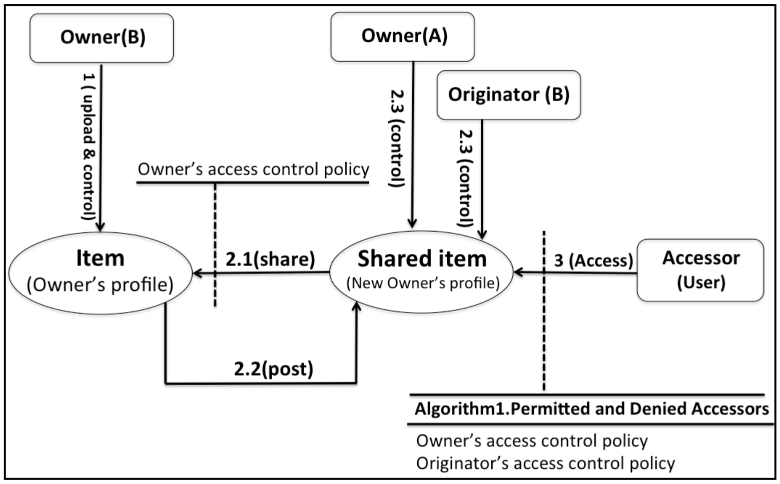


Figure 10: Simple sharing others users' contents scenario.

The sharing others users' contents scenario has complex cases. The first is when a user desires to disseminate content that has tagged users who have already been tagged by the owner of this content. Thus, there are stakeholders who have to be involved in the process of making collaborative privacy decisions over who can repost the content in her/his profile. For example, suppose Bob (B), who is the owner in our prior example, has a photo where Carol (C) and Dave (D) are explicitly identified through tags. When Alice (A) desires to share this photo with her social network, her request should satisfy the owner's and stakeholders' privacy requirements. We are going to discuss how this works in Chapter 4 by introducing the AccessorSharing algorithm. Assume the result of the collaborative decision grants Alice permission to share the photo with her social network. The photo will be in turn posted in her profile; consequently, it is collaboratively controlled by Alice, who is a new owner of the reposted photo, and Bob who represents the original controller of this photo. Our PermittedandDeniedAccessor algorithm, which we will introduce in the next Chapter, offers the process to make the cooperative decision over who can view the reposted item in the new profile. However, in existing OSNs the decision about sharing Bob's content and the access restrictions over who can see the reposted photo in Alice's profile are only regulated by the owner of the profile, regardless of the privacy requirements of the other connected users such as Carol and Dave in our example. Figure 11 shows our analysis for the first complex case of sharing others users' contents scenario.

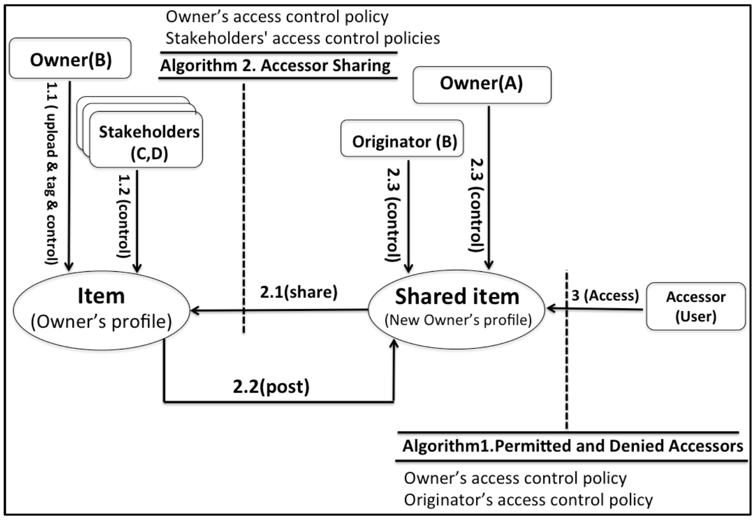

Owner's access control policy
Stakeholders' access control policies
Algorithm 2. Accessor Sharing
Owner(B)
1.1 (upload & tag & control)
Stakeholders (C,D)
1.2 (control)
Owner(A)
Originator (B)
2.3 (control)
2.3 (control)
Item
(Owner's profile)
2.1(share)
Shared item
(New Owner's profile)
3 (Access)
Accessor (User)
2.2(post)
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Originator's access control policy

Figure 11: Tagging- Sharing others users' contents scenario.

The last case in the sharing others users' contents scenario is more intricate. When Alice (A) desires to share a photo that original was posted by Edward (E) in Bob's profile and he (Edward) tagged Carol (C) and Dave (D) in the photo, those users who are linked to this photo in diverse ways should participate in the process of making a collaborative decision about Alice's sharing request. As we mentioned before, the AccessorSharing algorithm will be discussed in Chapter 4 in order to enable associated users to make a collective privacy decision. However, existing privacy mechanisms in OSNs enable the owner of the profile to be the sole decision maker. Figure 12 demonstrates Posting-Tagging-Sharing others users' contents scenario that is generated by sharing content was uploaded by a contributor.

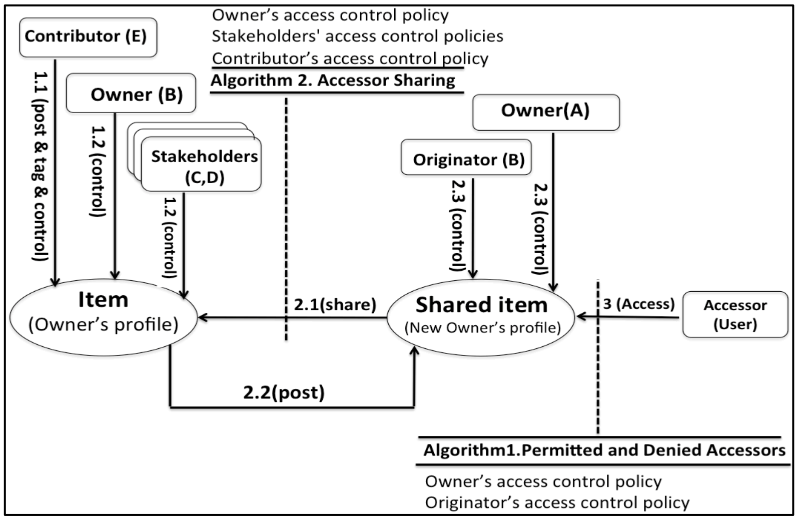

Owner's access control policy
Stakeholders' access control policies
Contributor's access control policy
Algorithm 2. Accessor Sharing
Contributor (E)
Owner (B)
Stakeholders (C,D)
1.1 (post & tag & control)
1.2 (control)
1.2 (control)
Owner(A)
Originator (B)
2.3 (control)
2.3 (control)
Item (Owner's profile)
2.1(share)
Shared item (New Owner's profile)
3 (Access)
Accessor (User)
2.2(post)
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Originator's access control policy

Figure 12: Posting-Tagging-Sharing others users' contents scenario.

3.3.3.3 Sharing other users' content and posting it in someone else's space

The last multiple controllers scenario is caused by the usage sharing feature in OSNs, when an intermediate user is involved in a sharing scenario. Let us consider, for example, situation where Alice (A) views a photo in Bob's (B) space and desires to share and repost the photo in Carol's (C) space, where Carol is one of her friends. Then, the photo will be in turn posted in Carol's space, who thus becomes the new owner of Bob's photo. We call Bob the originator and Alice the contributor who reposts the photo. Analyzing the relationship between connected users and shared content allows us to develop a collaborative method for specifying privacy policies of shared content. This scenario will be covered in Chapter 4 by the PermittedandDeniedAccessors algorithm. However, in current OSNs policies over who can access the photo in Carol's profile are only created by Carol who is the owner; in contrast, the original owner, who is Bob in our example, will not be able to influence the access restrictions over the reposted photo. Also, Alice, who posted the photo in Carol's profile, cannot regulate her privacy requirements under existing privacy protection mechanisms. This simple case of sharing other users' content and posts it in someone else's space is presented in Figure 13.

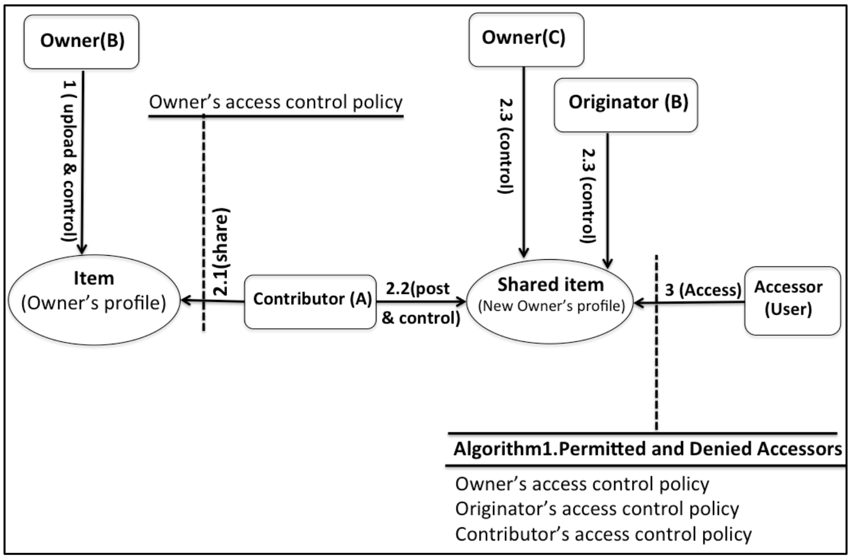


Figure 13: Simple sharing other users' content and posts it in someone else's space scenario.

A more complex situation occurs when the original content has tagged users. By following a prior example, suppose Bob's photo that Alice desires to share with her friend Carol is linked with Dave (D) and Edward (E) by tagging. We believe all connected users such as Bob, Dave and Edward have the right to participate in the access control process to restrict dissemination of the photo. For a photo in Bob's space, we consider him as the owner, Dave and Edward are stakeholders and Alice is a requester. When a collaborative decision grants Alice the permission to share the photo with her friend Carol, the access restrictions over who can view the photo in Carol profile should be cooperatively regulated by all users who are related to the considered resource. Our related users are classified as follows: Alice is the contributor, Bob is called the originator and Carol is the owner. However, unfortunately, current OSNs offer limited methods for specifying privacy policies in this scenario. In fact, it supports the owner of the profile, where the shared content resides, to be the sole decision maker for the sharing request, which in our example is Bob, and Carol for the accessing request. Figure 14 illustrates this tagging-sharing other users' content and posts it in someone else's space scenario and which algorithms, which will be discussed in more detail in the next Chapter, will be used to decide how collaborative users can participate in the access control decisions for shared contents.

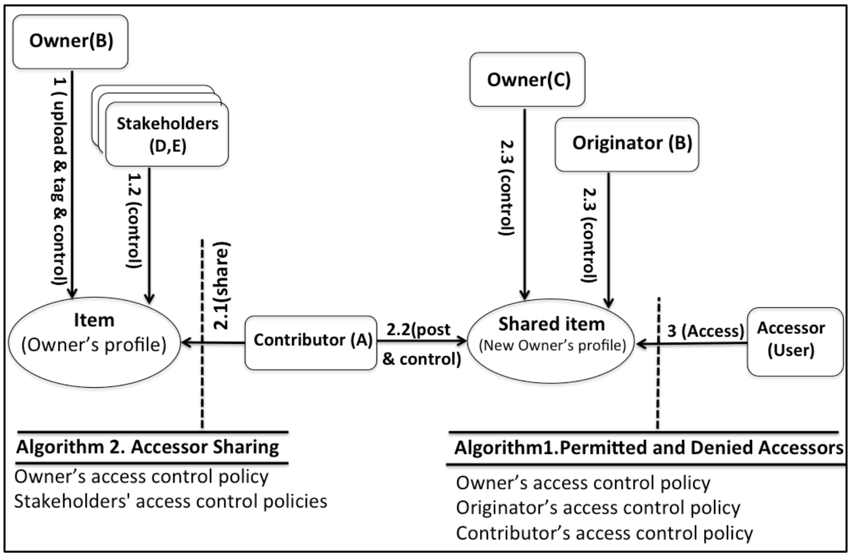

Owner(B)
1 (upload & tag & control)
Stakeholders (D,E)
1.2 (control)
2.1(share)
Item (Owner's profile)
Contributor (A)
2.2(post & control)
Owner(C)
2.3 (control)
Originator (B)
2.3 (control)
Shared item (New Owner's profile)
3 (Access)
Accessor (User)
Algorithm 2. Accessor Sharing
Owner's access control policy
Stakeholders' access control policies
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Originator's access control policy
Contributor's access control policy

Figure 14: Tagging-sharing other users' content and posts it in someone else's space scenario.

The most complex form occurs when Alice (A) decides to disseminate a photo that was originally uploaded into Bob's space by Frank (F) who also tagged Dave (D) and Edward (E) in the photo. Consequently, for the photo in Bob's space we call Alice the accessor who requests to share content, Frank the contributor of the content, and Dave and Edward are stakeholders of the photo. Our AccessorSharing algorithm supports all collaborative users to participate in the sharing control policy process over who can disseminate the shared content. Furthermore, when the photo is reposted in Carol's space, Alice becomes a contributor, Carol is the owner, and Bob is the originator. The collaborative access control polices of the photo in Carol's space, which is given by the result of the PermittedandDeniedAccessors algorithm, are regulated by all users who are related to the considered photo. The posting-tagging-sharing other users' content and posts it in someone else's space scenario is demonstrated by Figure 15. On the other hand, current OSNs offer only an individual process to regulate the access control policy for this scenario.

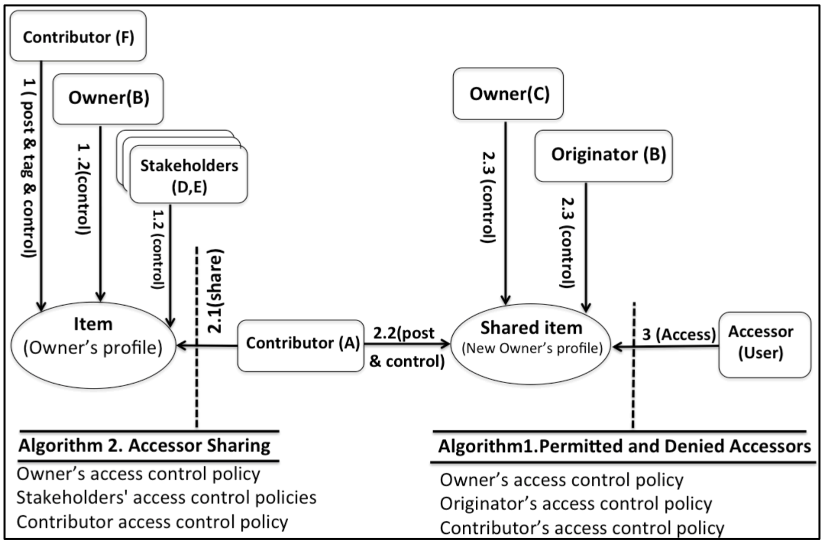

Contributor (F)
1 (post & tag & control)
Owner(B)
1. 2(control)
Stakeholders (D,E)
1.2 (control)
2.1(share)
Item (Owner's profile)
Contributor (A)
2.2(post & control)
Owner(C)
2.3 (control)
Originator (B)
2.3 (control)
Shared item (New Owner's profile)
3 (Access)
Accessor (User)
Algorithm 2. Accessor Sharing
Owner's access control policy
Stakeholders' access control policies
Contributor access control policy
Algorithm1.Permitted and Denied Accessors
Owner's access control policy
Originator's access control policy
Contributor's access control policy

Figure 15: Posting-tagging-sharing other users' content and posts it in someone else's space scenario.

To conclude, in this chapter we presented scenarios where contents are linked with multiple users who are explicitly identified through posts, shares, tags or other metadata. We believe to protect users' privacy in OSNs, we first need to analyze multiple controller scenarios to recognize users who have the right to participate in the privacy setting for a shared item. As we previously investigated, whenever types of sharing tools apply to an item in OSNs, multiple controller situations will occur. We have analyzed three main classifications of sharing patterns, profile sharing, relationship sharing and content sharing. We focused on the last pattern where content is the shared data. This pattern has three subcases that are tagging, posting and sharing. In general, existing OSNs provide limited access control that supports only a single user's privacy requirements in all the aforementioned multiple user scenarios. This limitation leads to disclosures of other individual user's information of which they may be unaware. In particular, they may be unaware of the fact that their content is disseminated and controlled by someone else, and that they cannot influence the privacy setting applied to their data. By requiring the approval from associated controllers, we offer access control models that enable collective users to manage and control their shared contents' access and dissemination policies. Our collaborative access control models are represented by the

PermittedandDeniedAccessors, Controller Sharing and AccessorSharing algorithms, which will be discussed in more detail in the next Chapter.

Chapter 4

4 A Collaborative Access Control model

Current OSNs offer access control mechanisms where the decision over who can access or disseminate the shared item is solely regulated by the owner of the profile where the considered item is. OSNs are multi-user virtual environments, as mentioned in Chapter 3, that raise several cases where contents can have multiple controllers. However, these multiple controllers are not allowed to mange and control their data in all developed OSNs, which provide only the most basic access control mechanism. Hence, a lack of collective management can lead to de facto public disclosure. In addressing user privacy, our intent is to let each user in a set of collaborative controllers identify her/his privacy requirements and participate in the process of making a collaborative access control policy. In these circumstances, each associated controller might have different and possibly contrasting privacy requirements. In response to this, in this Chapter, we present an approach that combines different controllers' privacy preferences into a single privacy policy. In particular, how to resolve such conflicts in contradictory privacy requirements and to support a reasonable access control model to make a collective decision over shared contents in OSNs is an open problem. Additionally, it is unclear how to combine all the privacy requirements for a shared item without violating individuals' requirements. As a consequence, the purpose of our approach is not only to combine the multiple controllers' privacy settings, but take into account various factors such as types of relationships, controllers' types and weights, the distance between users, accessors' weights, and a trust inference algorithm to establish an effective methodology for making collaborative access control decisions that achieves an optimal balance between availability and protection of shared items in OSNs.

We begin with an abstract representation of an OSN and formal definition of our access control model. Moreover, we present a collaborative policy specification scheme for access control and authorization administration. Afterward, we discuss our principles for a conflict resolution strategy. Finally, based on these considerations, we introduce our algorithms for collective privacy management, where the first algorithm, called PermittedandDeniedAccessors, produces the final lists of accessors who are permitted to view the shared data and those who are denied. We also introduce the AccessorSharing and ControllerSharing algorithms to reach collaborative decisions about who can disseminate the shared item.

4.1 Representation of OSNs

In this Section, we provide an abstract representation of an OSN. Our purpose is not to represent any concrete OSN, but to specify the significant elements of OSNs, upon which to construct our collaborative access control model. In addition, we introduce our scheme for expressing privacy policies for a collaborative access control model in OSN.

In the beginning, an OSN is a relationship network, a set of users, a set of contents and a set of user relationships. The relationship network of an OSN is an undirected and labeled graph where nodes denote users, edges represent their interdependencies and the label indicates the type of the relationship between users such as family, friend, colleague, coworker, etc. Current OSNs, offer certain and fixed types of relationships, for example the followee-follower relationship or friends relationship. Indeed, the privacy risk of lack of support for the varied types of relationships in existing OSN sites is studied by several researchers such as Gates [2007]. It is clear that users should be provided with more flexible relationship-based access control to govern access to their information, especially when users collaboratively create content. As in the real world, people can have in mind a specific audience for accessing, sharing or disseminating their pictures, events or activates. In order to have flexible relationship-based access control, our model supports varied types of relationships and enables users to specify accessors based on their relationship type. Moreover, in reality, OSNs have an important feature that enables a user to selectively share content with a specific group of people. This is called ‘circles’ in Google+, a social networking service introduced in 2011. Similarly, ‘lists’ or ‘groups’ are available in networks such as Facebook or Twitter. It is a useful mechanism allowing users to organize their networks and to effortlessly find users or friends, who may have the same hobbies, interests, schools, political standpoint etc. OSNs offer for each user a web space, which is called a wall or profile in Facebook, where they can define personal information, a list of contacts and their photos and albums and customize it as they wish. Bedside the relationship network, which is usually used to model OSNs, we also employ trust a social network. Similarly for the relationship network, trust relations between users in OSNs can be modeled as graphs. A trust social network, is modeled using a directed labeled graph where each node represents a user, edges denote the trust relation and labels indicate the trust value of the relationship. A trust value assigned to each edge expresses how much one user trusts another.

4.1.1 Controllers' definitions

We defined four types of controllers in OSNs, accommodating the special ownership requirements coming from multiple associated users scenarios for managing the shared items collaboratively as we discussed and investigated in the previous Chapter. The idea of classifying collaborative controllers is to identify associated users on the basis of their relationships with a particular item (shared item). They are the owner, who owns the data item in his/her profile, a contributor, stakeholder and originator. In our approach, we aim to cooperatively employ their privacy requirements in collaborative access control governing shared contents. We formally define these controllers as follows:

Definition (1) Owner: Let *I* be a data item in the profile of user *u* in an OSN. The user *u* is called the owner of *I*. In addition, the owner could be a user who owns the profile where a shared item is in turn posted.

Definition (2) Contributor: Let *I* be a data item in the profile of someone and user *u* be a user who shares *I* with her/his social network or another user's social network. Moreover, *u* could be a user who publishes *I* in some else's profile. The user *u* is called the contributor of *I*.

Definition (3) Stakeholder: Let *I* be a data item in the profile of any user in an OSN and *T* be the set of tagged users linked with *I*. A user *u* is called a stakeholder of *I* if $u \in T$ and is not the owner of *I*

Definition (4) Originator: Let *I* be a data item disseminated by user *x* from user *u*'s profile to *x*'s profile in an OSN. We call *u* an originator of *I* when it turns up in *x*'s profile. In other words, the originator is the user who owns the profile where the shared item first appeared.

4.1.2 The formal definition of the model

An OSN is characterized by the following core components:

- U = {u_1 , …, u_n} is set of users in an OSN such that each user has unique identifier.
- RT is a set of relationship types in OSN, which is a relationship network. Users connect to each other by various types of relationships in OSNs.
- R = {r_1 , …, r_n} is a list of relationship sets in OSN, and the relationship list of user i is r_i = {rl_{i1} , …, rl_{in}}, where $i \in$ U. Each entry of a relationship set is denoted by rl_{ij} = < u_j, rt_{ij} > where the first element is the user identifier $u_j \in$ U, and $rt_{ij} \in$ RT is the type of relationship between u_i and u_j.
- G = {g_1 , …, g_n} is a set of groups in OSN where each one has unique identifier. The set of user i's groups is g_i ={ug_{i1} , …, ug_{in}} ,where $i \in$ U. Each group, say $ug_{ij,}$ has a set of users who are its members are {um_{1j} , …. ,um_{nj}},where each $um_{ij} \in$ U.
- CT = {OW, CB, ST, OG} is a list of ownership types, where OW, CB, ST, and OG denote, respectively, the owner, contributor, stakeholder, and originator types.
- D = {d_1 , …, d_n } is a collection of users' data in OSN, and each element is a set of a particular user u's contents d_u = {c_{u1} , …, c_{un}} where $u \in$ U. Then, each c_{uj} is represented as <id_j, $content_j$ > where id_j is the unique item id and $content_j$ is the content of this item.
- *relation-members*: a mapping function applied to each user *u* to identify the set of users with whom he/she has a relationship rt; it is denoted by $u \overset{RT}{\rightarrow} U$
- *R.of.R*-members: a mapping function applied to each user *u* to identify the set of users who have a transitive relation of relationship type rt with u. It is denoted by $u \xrightarrow{RT.of.RT} U$.

- *controllers*: a mapping function applied to each content c to identify the set of users who have any type of ownership with it; it is denoted by $c \overset{CT}{\rightarrow} cU$.
- *administrated-groups* : a mapping function applied to each user to identify the set of groups that belong to her/him, denoted by $adminby(u) \rightarrow \mathrm{g}_u$.
- *group-members* : a mapping function applied to each group g to identify the set of users who are members in this group; it is denoted by $members(\mathrm{g}_i) \rightarrow \{\mathrm{um}_{1i}, \ldots, \mathrm{um}_{ni}\}$

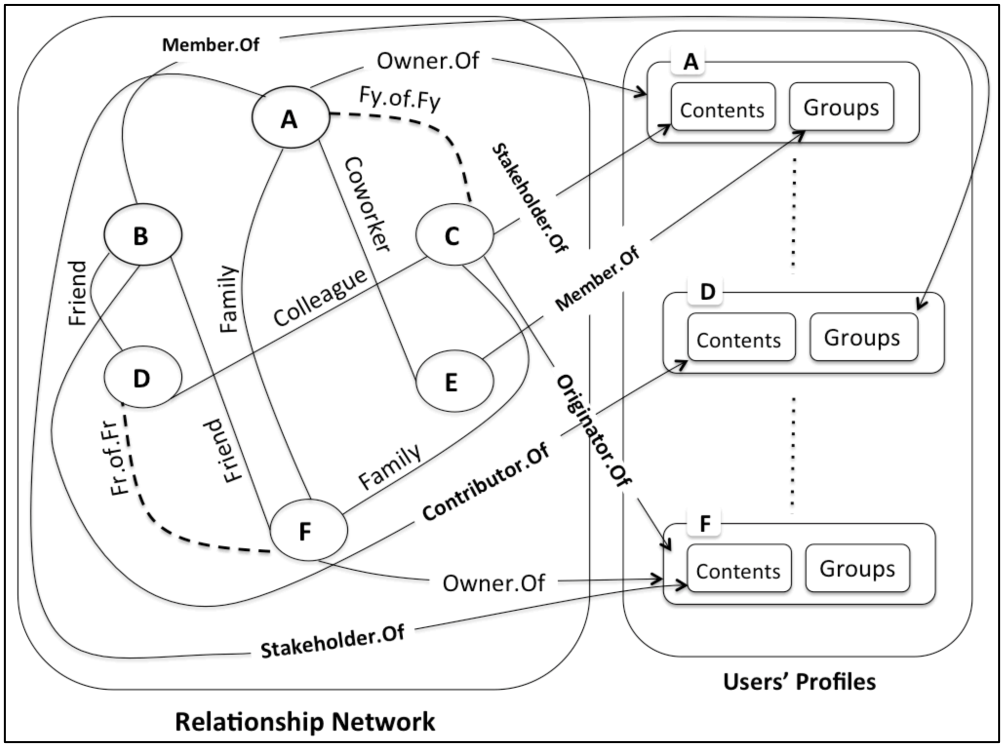


Figure 16: Representations of the social network structure.

Figure 16 illustrates an example of a graph representation of a social network used for exploring collaboration for social network features. It shows the relationships of six users, Alice (A), Bob (B), Carol (C), Dave (D) , Edward(E) and Frank (F), along with their ownership of contents and their memberships in groups of other users. Note, we work with a social network that structurally has undirected relationships. For example, if Alice has a coworker relationship with Edward, then automatically Edward has a coworker relationship with Alice. Also, different relationship types connect users, such as family, colleagues and coworkers in this example rather than just friendship, which is typically used to represent social relationships. Moreover, two users might have a transitive relationship which urges them to share and disseminate data in OSNs. For example, in Figure 16, Alice has a family relationship with Frank; similarly, he has a family relationship with Carol. Consequently, Alice and Carol have undirected transitive family relationship. In addition, each user can categorize her/his relationship list into groups; then, she/he is considered as the administrator of those groups. In this case, the group has one owner, who owns the profile where this group belongs, and has one or more members. For example, as we can see in Figure 16 Alice owns the group where Edward has a membership relation with her group.

Although groups are controlled and regulated by a single owner, contents may have multiple controllers. For instance, one of Frank's items has two controllers: the owner is Frank whose profile holds this item, and the originator of it is Alice. Furthermore, a user can control different items by varied types of ownership. An example of controlling multiple contents with diverse rights is that Carol is the originator of some item of Frank's content; on the other hand, she regulates one of Alice's contents as a stakeholder.

4.1.3 Privacy policy specification

Various access control schemes have been proposed recently to support fine-grained privacy policy specification for OSNs (e.g. [Carminati, et al. 2006, 2009b, Villegas, et al. 2008, Carminati, et al. 2011, Fong. 2011a]). However, those schemes support only a single controller to specify the access control policy of shared items. By supporting collective access control policies, sharing and interconnection among users in OSNs will be promoted in a more trustworthy environment. Before introducing the collaborative access control policy enforcement, we are going to introduce the specification of individual access control policies where a set of collective users, who are related to the considered data item, separately identify their privacy requirements.

4.1.3.1 Accessor Specification:

To enable the controller to accurately identify a set of users who can access her/his data and who cannot, we divide the accessor specification into two sets: accessor.permit and accessor.deny. In addition, for more precise specification, each set of accessors can be specified by three parameters that are user names, group names and relationship types. Accessor.permit is a list of users who are granted to access to the shared data by an individual controller. In contrast, accessor.deny is a list of users who are disallowed access to the shared items through user name, group name, or relationship name. Those represent types of accessors that a controller can customize to regulate her/his access control policy. We define the accessor specification as follows:

Definition (5) accessor specification : ac ∈ U ∪ G ∪ RT which can be user, group, or relationship type, where

- $U = \{u_1, \dots, u_n\}$ is a set of users who have any direct relationship with the controller.
- G is derived from a list of the controller's groups $g_i = \{ug_{i1}, \dots, ug_{in}\}$ where $g_i \in G$ and $i \in U$. For details see Section 4.1.2.
- RT indicates the relationship types where controller has them in her/his social network. In addition, controllers can use types of transitive relationships to specify who is allowed or denied access to her/his data. For details see Section 4.1.2.

Then, act ∈ UN ∪ GN ∪ RN is a list of accessor types, where UN, GN, and RN denote respectively, user name, group name, and relationship name. The permitted accessor specification is defined as a set, *permitted accessors* = {*permitted accessor* $_1$, …, *permitted accessor* $_n$ }, where each component is a pair <ac, act>. In addition, the denied accessor specification is expressed as a set, *denied accessors* = {*denied accessor* $_1$, …, *denied accessor* $_n$ }, where each element is of similar structure, <ac, act>.

When a subject is both permitted and prohibited to perform an action on an object, conflicts may occur. The combined use of positive and negative authorizations in our accessor specification brings conflict problems of how the two specifications should be treated. Although conflict resolution is a more complex matter and does not usually have a unique answer, we provide rules to regulate accessor.permit (*a.p*) and accessor.deny(*a.d*) sets properly as follows:

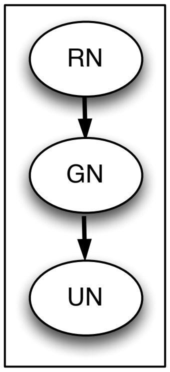


Figure 17: A taxonomy of accessor types.

From Figure 17, we can infer that the list of accessor types is a hierarchical list for classifying and identifying accessors. The taxonomy is from most the generic accessor type to most the specific accessor type; thus a relationship name is more general than a group name and a group is more general than an individual user. First of all, accessor.deny can be null. The controller allows everyone in his/her social network to access the considered shared data. A user's social network includes all users who have an existing relationship with the controller or a relationship of degree two from the controller (e.g. *friend.of.friend)*.

Secondly in a hierarchical list of accessor types, the intersection at the same level should be empty, where $a.p_i = \{a.p_{i1}, \ldots, a.p_{in}\} \cap a.d_i = \{a.d_{i1}, \ldots, a.d_{in}\} = \emptyset$,where $i \in U$, *ac* of each $a.p_{ij}$ and $a.d_{ij} \in RT$ and *act* of each $a.p_{ij}$ and $a.d_{ij}$ is RN. Likewise at the group name level, $a.p_i = \{a.p_{i1}, \ldots, a.p_{in}\} \cap a.d_i = \{a.d_{i1}, \ldots, a.d_{in}\} = \emptyset$, where $i \in U$, *ac* of each $a.p_{ij}$ and $a.d_{ij} \in G$ and *act* of each $a.p_{ij}$ and $a.d_{ij}$ is GN. Similarly for user name level $a.p_i = \{a.p_{i1}, \ldots, a.p_{in}\} \cap a.d_i = \{a.d_{i1}, \ldots, a.d_{in}\} = \emptyset$, where $i \in U$, *ac* of each $a.p_{ij}$ and $a.d_{ij} \in$ U and *act* of each $a.p_{ij}$ and $a.d_{ij}$ is UN. In OSNs, a user can be a member in multiple controllers' groups; therefore user could belong to denied group and permitted group at the same time. For this reason, in our model when a controller specifies their accessors, we investigate if there is a conflict in accessor specification that is raised by conflicting memberships. To resolve this type of conflict, we take into account the denied vote. For example, suppose Alice has a friend relationship type with Bob who is a member in Alice's coworker group and colleague group. When she permits her coworker group to access item and denies her colleague group to view this item, the system will deny Bob to access. Consequently, if we have one vote to authorize an access, and another to deny it, we apply the denial- takes-precedence principle [di Vimercati, et al. 2005]. The case of conflicting memberships may be more complex when we have an accessor who is a member in two permitted groups, but she/he is a member in one denied group. Accordingly, to solve such a conflict, we apply many-takes-precedence principle that is the higher number of positive/negative policies prevail over fewer positive/negative ones. As a result, *denial takes precedence* and *many takes precedence* are the conflict resolution policies we adopt to solve the conflicting memberships that may occur in the accessor specification step.[1]

The last rule in the accessor specification is resolving conflicts between UN, GN and RN by taking into account the *most-specific-takes-precedence* principle [di Vimercati, et al. 2007, 2005]. Based on our taxonomy of accessor types in the accessor specification, we consider UN the most specific policies and RN is the least specific. For clarification, consider the scenario where Alice permits her coworker group (GN) to access the shared item and denies her friends (RN). Bob has a friendship relation with Alice and at the same time is member in her coworker group. Thus, to solve such an accessor specification policy conflict, our model considers Bob as a permitted user. Let us discuss another case to illuminate how such conflicts can be solved by the *most-specific-takes-precedence* principle. Suppose Alice regulates her accessors policy as follows, deny family group (GN) and family relationship (RN) to access the shared item and permit Carol by name (UN) who has family relationship with Alice, so she is also a member in Alice's family group. According to our conflict resolution policy, we consider

[1] We use denial-takes-precedence here, but later, when considering multiple controllers, we will use permit-takes-precedence.

Carol as a permitted user even though she is denied in RN and GU accessor types because Alice permits her in the most specific type, which is UN.

In summary, our accessor specification supports both positive and negative authorizations which can cause conflicts when a controller specifies whose can access her/his data. Consequently, we apply different conflict resolution policies and some rules to solve such conflicts.

4.1.3.2 Data Specification:

The user profile, user's relationships and user's content embody the user's data in OSNs. However, our model focuses in the last component, the user's content, and assigns a level of sensitivity to content based on how much a disclosure would harm the user. Furthermore, sensitivity levels of shared contents help effectively to solve conflict between controllers. While the users should specify their sensitivity level of shared items in previous work as numbers [Hu, et al. 2013], people logically use linguistic expressions when they are asked about their sensitivity level of data. Consequently, we use fuzzy logic, which seems closer to the way users would express their tolerance, such as low, medium or high and so on. Sensitivity levels (*sl*) are assigned to the shared item by each controller who is related to the considered item. Hence, we introduce *sl* with varying degrees of sensitivity. Data specification supports the linguistic terms as a sensitivity level of a particular item for a controller. The linguistic variables, which are the input that are assigned by controllers, and numerical values, which are used in our algorithms which will be given at the end of this chapter, are defined as follows:

Table 1: Sensitivity levels

Linguistic term	Numerical value
NONE	0.00
LOW	0.25
MEDIUM	0.50
HIGH	0.75
HIGHEST	1.00

The definition of the data specification is as follows:

Definition (6) data specification: c ∈ D is a data item and sl is rational number in the range [0,1] which is assigned to c. The data specification is defined as a pair: (c, sl) where sl represents the sensitivity level.

4.1.3.3 Individual access control policy:

An individual AC policy consists of five elements where each controller is required to determine the list of permitted users as well denied users to the considered item. Moreover, an individual access control policy requires a controller, during policy specification, to assign a sensitivity level to the considered item. Formally, the Individual access control policy is $\boldsymbol{p} = < controller, \boldsymbol{TypeC}, \boldsymbol{shared\ item}, \boldsymbol{permitted\ accessors}, \boldsymbol{denied\ accessors} >$, where

- Controller is the user who regulates access policy over the considered item, where the controller ∈ U
- TypeC is type of the controller, where TypeC ∈ CT
- $shared\ item$ is denoted with a data specification defined in Definition 6
- $permitted\ accessors$ is a list of users who are allowed to access the item, represented with the accessor specification given in Definition 5
- $denied\ accessors$ is the list of users who are rejected to access the item, represented with the accessor specification given in Definition 5

To illustrate the aforementioned individual access control policy, we next turn to some examples as follows (in natural language):

1. Bob grants permission to any user who has a family relationship with him to view his travel photo identified by *travel1* with a high sensitivity level, where he is the owner of the photo.
2. Alice authorizes users who are members in her Co-project group or have coworker relationship type with her to view a photo that she posts it in her manager space identified as *schedul-meeting02* with a low sensitive level from her side, and then Alice is considered to be the contributor of the photo.

3. Dave prevents his friends to see his post; on the other hand, he allows his brothers, Bob and Frank, to view this post that is known as *invitation-BD20* with a medium level of sensitivity, where Dave is the owner of the post.
4. Edward denies his family members and users who are his coworkers to access a photo, *fun-event00*, that he is tagged in with a highest sensitivity level. Nevertheless, he authorizes users who are his colleagues and Carol to view the considered photo, where Edward is a stakeholder of the *fun-event00* photo.
5. Carol denies users who are her colleagues to view the post *writing-memory10*, that she allowed her sister Alice to share with her social network with a medium sensitivity level. Carol is considered as the originator of the shared post.

Are defined as follow:

1. *p* = < Bob, OW, < *travel1*, 0.75>, {<family, RN>}, {}>
2. *p* = < Alice, CB, < *schedul-meeting02*, 0.25>, {<coworker, RN>, < Co-project, GN>}, {}>
3. *p* = < Dave, OW, < *invitation-BD20*, 0.50>, {<Bob, UN>, <Frank, UN>}, {<friend, RN>}>
4. *p* = < Edward, ST, < *fun-event00*, 1.00>, {< colleague, RN>, <Carol, UN>}, {<family, RN>, <coworker, RN>}>
5. *p* = < Carol, OG, < *writing-memory10*, 0.50>, {},{<colleague, RN>}>

4.2 Requirements for a conflict resolution strategy

By identifying an individual access control policy for each controller belonging to a set of collaborative users that we have identified on the basis of their relationships with the considered content, now we need to turn to an aggregation process to yield a final access control policy for the shared content. However, before reaching the collaborative access control policy that aims to satisfy all associated controllers' desires, we need to define some rules and principles. The purpose of this step is to enforce the combining process of individual access control policies effectively. Indeed, when we try to combine diverse individual access control polices, which come from different perspectives and requirements, conflicts between controllers' policies are bound to happen. Conflict resolution is a complex problem and does not normally have a unique answer [Jajodia, et al. 2001, Lunt. 1989]. Nevertheless, the conflict issue can be addressed by defining a conflict resolution strategy. Varied criteria could be adopted to correspond different policies that enable the system to solve the conflict. A conflict resolution strategy consists of a set of rules that enables a system to decide either to permit or deny an accessor. The result of a collaborative access control policy depends on the chosen conflict resolution strategy. Therefore, in this section, we are going to explain our rules and principles to resolve conflicts. Then, those principles will be merged to work together to obtain the final lists of permitted and denied accessors of particular shared content from all the individual policies.

The tradeoff between providing privacy protection and the value of sharing in OSNs is a primary focus of our model. Consequently, the rules of our conflict resolution strategy are basically chosen to achieve the desired equilibrium between the privacy of online users and the goal of sharing data. The first principle is that, although all associated controllers should be able to control shared data, the impact and priority level of their policies is different. We call the impact of their policies the Controller Weight that is determined by the relationship between the controller and the shared item. In addition, as we discussed previously in Chapter 2, communication in a social network reflects human social interactions; hence, we believe connecting and sharing with other users in OSNs are not possible without trust. In fact, trust is critical to establishing any communication in OSNs. For this reason, we consider the trust value among the set of collaborative controllers and accessors as a second principle to resolve a conflict. Lastly, according to the most-specific-takes-precedence principle, we have the third principle which is Assessor Weight. For more details on our principles for the conflict resolution strategy, next we have a separate discussion section for each principle.

4.2.1 **Principle 1.** (Controllers’ Weight Scheme)

The essential step to resolve conflicts constructively is by assigning a weight (or score) to controllers, according to their categories. Even though all related controllers should have the ability to be involved and have an impact on making a collaborative access control decision over the shared item, we believe that each controller’s privacy requirements have distinctly different priority and influence. Because of this, we provide a weighting scheme that is required to assign a value to each connected controller. As we previously mentioned, there are four types of ownership: owner (OW), contributor (CB), stakeholder (ST), and originator (OG), where the idea behind those types is to identify collaborative users on the basis of their relationships with a particular shared item. Hence, a controller's weight is mainly calculated based on her/his relationship with the shared content. The weight is a rational number in the range [0,1], where the weight of controller *u* is denoted by *w* (*u*). According to the aforementioned content sharing patterns, we can infer the users who belong to a set of collaborative controllers of an item that has multiple controllers, and assign a weight to each of them.

In the tagging scenario in Figure 5, where we have the owner who uploads a picture to her/his profile and tags other users in it, we assign a weight of 1 to the owner. Additionally, we have stakeholders, a second type of ownership in the tagging situation, who are tagged users. Indeed, the owner is not required to ask for the permission of the stakeholders appearing in the picture, even if they are explicitly identified through tags. Hence, we believe stakeholders should be given the same owner priority to manage their identity through the photos across many audiences and people in their social networks. Similar to the owner, we assign a weight of 1 to the stakeholders.

Secondly the posting scenario, in Figure 6, in addition to the owner and stakeholders, we have the contributor who posts the content in the owner's space and might tag other users in the considered content. Here there are two circumstances to determine the contributor's weight. First, if a contributor belongs to the set of users who have any type of relationship or transitive relationship with the owner, we assign a weight 0.50 to the contributor. Otherwise, when the contributor does not belong to the owner's social network, which includes all users who have existing relationships or transitive relationships with the owner, we assign a weight of 0.25 to the contributor. In other words, a contributor's weight is based loosely on the distance between the contributor and the owner. If the contributor is connected with the owner by shorter distance (1 or 2), they get a higher weight than others. The reason behind our way of weighting the contributor is that when she/he is one or two degrees of separation from the owner, the possibility to have common users between them is high. Consequently, the number of accessors who belong to a contributor’s social network and can access, tag and share the item with others is high; thus, intuitively a contributor should have high weight to impact highly in the process of making collaborative access control decisions over the considered item. Our weighting approach is based on the confirmed result in sociology that friends tend to be similar [Feld. 1981, Carley. 1991]. A contributor who is in one degree of separation from the owner should have the most friends in common, while a contributor who is two degrees of separation from the owner should have fewer common friends and so on. To summarize, a contributor who belongs to the owner’s social network is weighted 0.50, otherwise 0.25. An exception is if the contributor posts a picture in the owner’s profile and tags her/himself and other users in a given picture; we consider her/him as a stakeholder because she/he is identified through the tag.

The sharing scenario has three subcases; first, in Figure 7, when the owner or one of the stakeholders shares their contents with any user in the OSN, the item will be reposted in the user’s space. For reposted content, we assign a weight of 1 to the owner of the profile where the content has been moved to; moreover, we have an originator (the initial owner or stakeholder) who shared the considered content with new owner. The approach we are going to apply for the originator’s weight is to have it equal to the contributor’s weight, which was previously discussed. The originator who belongs to the new owner’s social network is weighted 0.50, otherwise 0.25.

In the second sharing situation, in Figure 11, when a permitted accessor, one who is allowed by collaborative access control to view an item, desires to share the item with her/his social network, she/he becomes the owner after reposting the considered item in her/his profile. Then, this owner of the reposted item is weighted 1 as the owners were in the previous scenario. Furthermore, the owner of the initial item is converted to being the originator of the reposted item. To assign a weight to the originator, we consider a trust value that is a new factor in place of the prior factor, which was the degree of separation between two users. Trust in social networks has been generally discussed in the previous Chapter, but in the next section we are going to precisely illustrate a trust inference approach that we adopt to determine how much a user in the trust network should trust another user who is not directly connected to her/him. To weight the originator in this scenario, first we infer how much the originator trusts the owner of the reposted item’s profile, and then subtract from one the trust value that is computed. Finally, we get the originator's weight as follows:

$$w(OG) = 1 - TG.infer(OG, OW) \tag{1}$$

$$\textbf{if} \quad TG.infer(OG,OW) = 1 \ \textbf{then}$$

$$w(OG) = 0.25$$

where TG is trust graph and $infer$ is a function to infer how much the originator trusts the owner according to the trust graph. The trust network and inferred trust value will be discussed in more detail in Section 4.2.3. However, here we focus on weighting controllers. Based on equation (1), if the originator trusts the owner at the highest level which equals 1, the originator's weigh will be *w(OG)*= 0. In this case, we weigh the originator with minimum controller weight in our model which equals 0.25.

The last case of usage sharing in OSNs occurs when an intermediate user is involved in the sharing scenario. For disseminated content, as in Figure 13, we assign weight 1 to the owner. To weight a contributor, who is the intermediary, we adopt a similar approach that we used to weight contributors in the previous scenarios. There are two cases to consider for the originator. First, if the originator has a relationship with the owner of the profile where the item will be reposted, we apply the same approach that we adopted to weight the contributor in the second scenario. On the other hand, if the originator does not know the owner of the profile where the content will be disseminated, we weight her/him by using the trust inference method as shows in equation (1).

In brief, the Controllers' Weight scheme is a method to determine a priority and impact level of a controller's policy. Although all associated controllers should be allowed to define their access control policies over a shared item, we believe it might be more effective to assign weights to collaborative controllers. Hence, they have a different priority especially when they participate in the process of making a collaborative decision over a shared item. Our Controllers' Weight scheme is summarized as follows:

Table 2: Controllers' Weights.

Controller Type	Status	Weight
Owner	All	1
Stakeholder	All	1
Contributor	When (simple, accessor sharing by intermediary) If (Distance = 1 or 2)	0.50
Contributor	When (simple, accessor sharing by intermediary) If (Distance >=3)	0.25
Originator	When (controller sharing) If (Distance = 1 or 2)	0.50
Originator	When (controller sharing) If (Distance >=3)	0.25
Originator	When (accessor sharing)	$w(OG) = 1 - TG.infer(OG, OW)$
Originator	When (accessor sharing by intermediary who has relationship with originator) If (Distance = 1 or 2)	0.50
Originator	When (accessor sharing by intermediary who has relationship with originator) If (Distance >=3)	0.25
Originator	When (accessor sharing by intermediary who does not have relationship with originator)	$w(OG) = 1 - TG.infer(OG, OW)$

4.2.2 **Principle 2.** (Accessors' Weight Scheme)

An individual access control policy (p) holds positive and negative authorizations. Also, those authorizations are represented by the accessor specification, which has been identified earlier in Definition 5. We believe that not the all accessors are equal because they are specified differently. Thus, we assign diverse weights to accessors based on how they are authorized or denied. In response to this assumption, we adopt the *most-specific-takes-precedence* principle to weight our accessors [di Vimercati, et al. 2005, 2007]. Actually, we weight the accessor by weighting her/his specification type *w(act)*,where *w* is a rational number in the range [0,1] and act is a list of accessor types as identified in Definition 5. As we discussed above in the accessor specification section, accessors can be granted or denied by the types of accessor specification that can be a relation name, group name or user name. Those are respectively ordered from most generic type to most specific type; consequently, user names, which is the most specific type, takes the highest weight that is defined as *w (act)*=1, where *act* ∈ UN. For a group name type, we assign a weight equal to 0.75 (high), i.e. *w (act)*= 0.75, where *act* ∈ GN. Finally, *w (act)*= 0.50 (medium), where *act* ∈ RN indicates a generic type of accessor specification.

4.2.3 **Principle 3.** (Inferring Fuzzy Trust)

Trust has a critical position in communications and interactions between people; consequently, social life is simply not possible without trust. In fact, OSNs are trying to simulate real social networks on the web. In real social life, our relationships with people can be classified with different circles, like family, friends, coworkers, colleagues, classmates, etc. Moreover, even in the same colleagues circle, we may stay closer to some people than to others. However, current OSNs offer a basic mechanism like a friend list that does not support a distinction between the types of friendships. We believe that additionally, OSNs should hold multiple types of relationships to connect users and begin transactions; it might be more controllable for OSN users if we distinguish the tie strength and the relationship quality and intensity between users. Thus, we adopt a trust relation to estimate the intensity of a relationship between users.

A graph structure is usually used to model trust relationships of users in OSNs. Modeling the users as nodes, trust relationships as directed edges and trust values as edge labels, this graph is called the trust graph. Figure 18 shows an example of a trust graph representation of a social network used for inferring trust values between users. Both the rust and relationships graphs, which was previously represented in Figure 16, are used in our collaborative access control model.

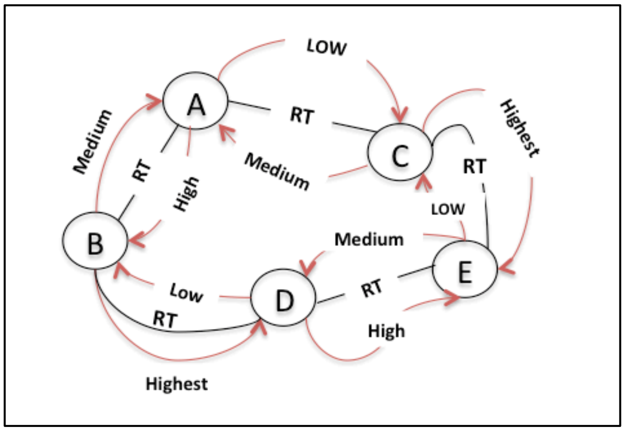


Figure 18: Trust Graph.

The trust graph is viewed as directed graph. Users explicitly identify a trust value for those with whom they have a direct relationship. When users are not directly connected, the process of determining how much the first user in the trust graph trusts another one is called trust inference. As mentioned in Chapter 2, various methods to infer trust have been proposed in the literature. Using fuzzy logic and its operators in trust models has been first considered in OSNs by Lesani and Bagheri[2006] and then by Lesani and Montazeri [2009] and Kim and Han[2009].We adopt the Lesani and Bagheri [2006] approach to calculate the trust value between two users in a trust social network who are not directly connected. The FuzzyTrust algorithm is similar to the TidalTrust algorithm[Golbeck. 2005], which was proposed by Golbeck for deriving a trust relationship between two users in the social network using the FOAF vocabulary, as it uses the same shortest path for trust computation. But with a slight difference, they propose fuzzy linguistic terms such as low, medium, medium low, medium high and high to denote trust of other users and develop an algorithm based on these rather than (1,2,3..n) in TidalTrust [Golbeck. 2005]. The FuzzyTrust algorithm considers the problem when there is conflict of information from different sources in a large social network. Furthermore, this algorithm calculates trust for indirect connections from stronger and shorter paths as follows:

- Iterate the nodes from trustor (source) to trustee (sink) similar to the breadth first search, level by level, to find the shortest paths through other users who are called recommenders.
- Set the path strength as a fuzzy set from trustor (source) to trustee (sink) through other users who are called recommenders.

We use the FuzzyTrust algorithm to infer trust values in our algorithms but with a slight difference in some considerations. Basically, a trust social network is raised when each user in an OSN gives individual trust value to users who have a direct connection with them, and a trust value can be a fuzzy linguistic expression such as low, medium, high and highest that are provided by our model *T* = {low, medium, high, highest}. The linguistic variables, which are the input that are assigned by users, and numerical values, which are used in our algorithms which will be given at the end of this chapter, are defined as follows:

Table 3: Trust values.

Linguistic term	Numerical value
LOW	0.25
MEDIUM	0.50
HIGH	0.75
HIGHEST	1.00

Trust values are .25, .50, .75or 1in our trust graph. We consider the assumption that users in OSNs who are directly connected to each other by any type of relationships should have at least the default level of trust which is Low. In other words, the existence of an undirected relationship between two users is associated with the existence of a directed trust relationship between those users. Furthermore, we consider the same characteristics of trust that the FuzzyTrust algorithm adopts, i.e. that it is asymmetric. Considering the asymmetry of trust, the trust social network would be a directed graph, and this property tells us that the trust level is not necessarily identical in the two directions between two users.

Our main motivation for selecting trust a FuzzyTrust algorithm to work with a trust graph for trust inference is so we can use a fuzzy linguistic expression. In OSNs we mainly deal with end-users, consequently those linguistic terms are easier and natural for users to assign trust values to others. Likewise end-users prefer to hear linguistic expressions when they ask others about their trust of an unknown user (trustee). In addition, we can handle conflicts where trust inference in a trust online social network can encounter contradictory information. Moreover, the results of Lesani's and Bagherip's simulation show that the FuzzyTrust algorithm offers more precise information than TidalTrust.

Finally, the main idea for incorporating trust into a set of principles we use to produce a collaborative access control is that it makes privacy very controllable for OSN users. Also, the estimation of the trust value is quite useful to identify privacy threats. Users' opinions in OSNs can be evaluated by taking trust relations into account.

4.3 Algorithms for Collaborative Privacy Decisions

As we have given definitions of the conflict resolution strategy principles. Also, we defined our policy specification scheme that has controller types, accessor specification and data specification. Now, based on these considerations, we propose to address the problem of collaborative privacy polices of shared items in OSNs. In this section, we investigate how collaborative privacy management of shared data with our principles can be achieved. Intuitively, this problem allows a conflict in the policies associated with controllers attempting to control their data. We develop a flexible and lightweight framework to achieve effective conflict resolution and to support the tradeoff between privacy and the benefit of sharing data in OSNs. The combined use of accessor weight, controller weight, and trust inference has been adopted as a convenient approach to resolve multiparty privacy conflicts in OSNs. This section is structured as follows. We begin by describing algorithms in pseudo code form then explain algorithms in detail and finish with examples.

4.3.1 PermittedandDeniedAccessors Algorithm

The proposed algorithm presented below tries to find the final lists of accessors who are permitted to view the shared data and those who are denied. The PermittedandDeniedAccessors algorithm is given as Algorithm 1.

Algorithm 1.PermittedandDeniedAccessors

input : $(item, P = \{< controller, TypeC, shared\ item, permitted\ accessor, denied\ accessor >\},$ TG) // *item* is a particular shared item where *item* ∈ D , *P* set of *p* where each *p* ∈ P and *p* is access control policy that is assigned by each associated controller for the data item in question and TG is trust graph where each user u ∈ U assigns a trust value to those who they have direct relationships with.

output : $final\ permitted\ accessor$: int [ ] , $final\ denied\ accessor$: int [ ]

var

```
  number-u-deny       : int       init 0
  number-u-permit     : int       init 0
  u-existence         : int       init 0
  decision- permit    :double     init 0
  decision- deny      :double     init 0
  list-users          : int [ ]   null
```

```
begin
   for each p in P  do
      add all user in  $permitted\ accessor$ and $denied\ accessor$ to list-users
   for each u ∈ list-users [ ] do // relevant user who is derived from $permitted\ accessor$ and
                                                 $denied\ accessor$
     {
        for each p in P  do
          {
              if  u ∈ $permitted\ accessor$ then
                number-u-permit ++
              else  u ∈ $denied\ accessor$ then
                  number-u-deny ++
          }
        u-existence= number-u-permit + number-u-deny // checking if there are no conflicts
        if  u-existence= number-u-permit then
          {
              add user u to $final\ permitted\ accessor$ [ ]
              remove user u from list users [ ]
           }
       else if  u-existence = number-u-deny then
          {
              add user u to $final\ denied\ accessor$[ ]
              remove user u from list users [ ]
           }
       else  // conflict case
         for each $controller$ who permites user u  do //$controller$ is a controller id from p
              decision- permit + = $\boldsymbol{w}(controller) * \boldsymbol{w}(u) * \boldsymbol{TG.infer}\ (controller \rightarrow u)\ *\ \boldsymbol{sl}\ (item)$
        for each $controller$ who denies user u  do
              decision- deny + = $\boldsymbol{w}(controller) * \boldsymbol{w}(u) * (1 - \boldsymbol{TG.infer}(controller \rightarrow u)\ )$
                                     $* \boldsymbol{sl}\ (\ item)$
         if  decision- permit $\geq$  decision- deny then
             add user u to $final\ permitted\ accessor$ [ ]
        else
32:          add user u to $final\ denied\ accesso$ [ ]
33:    } // end of users' loop
34:       return $final\ permitted\ accessor$ [ ] and $final\ denied\ accessor$ [ ]
```

In the above algorithm, two steps are performed to get the final permitted users collection and the denied users collection given several multiparty access control policies. The algorithm receives policy (p) from each controller associated with targeted item as input and produces two lists of accessors as output. Moreover, Algorithm 1 takes as input the trust graph (TG), where a label is assigned to each edge to indicate the trust value of the relationship, to infer trust values between users. $final\ permitted\ accessor$ is for users who are allowed to access the targeted item and $final\ denied\ accessor$ is the list of users who are denied to view the item. First of all, permitted and denied users from all relevant (p) are stored in *list-users* list*.* Then, each user (u) in *list-users* there is a loop to check the existence of this particular user (u) in $permitted\ accessor$ or $denied\ accessor$ of all controllers policies. From $permitted\ accessor$ and $denied\ accessor$ tuples in each controller's policy (p) we determine the number of permitted votes for a certain user (u) and the number of denied votes for the same user (u). These elements are represented by the accessor specification defined in Definition 5. If there is full consensus among associated controllers about permitting or denying a certain user to access the shared item, the algorithm can yield the final decision without moving to the conflict resolution step. To illustrate, assume Alice and Bob are co-controllers of particular item and David has an existing friendship with Alice as well as Bob. Both of them define their own access control policy (p). Alice's policy states that her friends can view this item. Bob's policy says that his friend David can view the shared item. Consequently, to yield the final decision about David's access, the algorithm aggregates Alice's and Bob's policies. As a result of collaborative privacy polices, David is allowed to view the targeted item. Moreover, regardless of how many controllers are associated with a targeted item, if the algorithm finds that the number of users existence (*u-existence*) equal to the number of user's permitted (*number-u-permit*) or the number of user's denied (*number-u-deny*), it can yield the final decision without moving to the conflict resolution step. To illustrate this shortcut, which is expressed in lines 14 to 23 in the algorithm, assume that Alice, Bob and Carol took a photo together. Then, they define their own access control policy as follows: Alice allows her family and denies her friends, Bob only denies his friends to access the photo and Carol denies her classmates. Suppose David is one of Alice's family members and he does not belong to Bob's friends or Carol's classmates; the final decision about his access will be permit because one of the linked controllers allows him and he does not belong to the other controllers' denied accessor list.

In fact, multiple controllers of shared items often have different privacy concerns over the data; thus, privacy policy conflicts can always exist when taking into account the collaborative control over the shared item. For this reason, the lines 24 to 33 of Algorithm 1 attempt to reach the final collaborative access control policy by resolving the privacy policy conflicts. The combined use of the aforementioned principles has been adopted in this part of the PermittedandDeniedAccessors algorithm. Recall that our goal is that each controller associated with a shared item has the ability to affect the final decision. To evaluate the access of a user who has conflicting policies, the decisions of all controllers, who permit the user to access, are aggregated to finalize the value of the permitted decision (*decision- permit*). On the other hand, the PermittedandDeniedAccessors algorithm aggregates the values of decisions (*decision-deny*) that are regulated by controllers who denied the user's access. The values of permitted decisions and denied decisions are computed with following equations:

$$\text{decision- permit} += \boldsymbol{w}(controller) * \boldsymbol{w}(\mathrm{u}) * \boldsymbol{TG.infer}(controller \rightarrow u) * \boldsymbol{sl}\,(item) \qquad (2)$$

$$\text{decision- deny} += \boldsymbol{w}(controller) * \boldsymbol{w}(u) * (1 - \boldsymbol{TG.infer}(controller \rightarrow u)) * \boldsymbol{sl}\,(item) \qquad (3)$$

where $\boldsymbol{w}(controller)$ is the weight of the controller derived from the Controllers' Weight Scheme and $\boldsymbol{w}(u)$ is the weight of the accessor that is calculated based on the Accessors' Weight Scheme, which is the third principle in our conflict resolution strategy. Also, we believe the trust value between each controller of a shared item and users who have conflicting policies about their access plays a critical role in making a collaborative decision. The trust value between controllers and accessor ($TG.infer(controller \rightarrow u)$) is inferred by the FuzzyTrust algorithm, which we discussed previously. The last element in the aggregation equations is the sensitivity level (sl) that reflects the controllers' privacy concern over the shared item, and is derived from the $shared\ item$ of a policy (p). The

$shared\ item$ element is represented by a data specification, which was defined in Definition 6. Then, the aggregated permitted decision value (*decision- permit*) is utilized as a threshold for making the final result. The final decision of our conflict resolution approach is then determined as follows:

$$\begin{array}{l} \text{decision- permit} \geq \ \text{decision - deny} = \text{permit} \\ \text{decision- permit} < \ \text{decision - deny} = \text{deny} \end{array} \qquad (4)$$

If the value of permitted decision is higher or equal than value of denied decision, the final decision is permit. Otherwise, the access of the user to the shared item is rejected.

A primary focus of our approach is based on a tradeoff between privacy and utility in OSNs. For this reason, we apply the aforementioned principles in our conflict resolution strategy. Indeed, whichever principles we consider to resolve conflicts that arise in a making collaborative decision, we will always find some situations for which they do not have a definitive scientific answer. Thus, the end result of our approach moves toward the permitted decision to solve the remaining conflicts. To see the PermittedandDeniedAccessors algorithm in action, let us consider the following example.

Example 1.

Suppose the scenario where Alice (A), Bob(B), Carol(C) and Edward (E) took a photo together. The photo is posted by Bob in Alice's profile and Carol and Edward are tagged in the photo. Therefore, according to our classifications of ownership, which are defined in Definitions 1, 2, 3 and 4, Alice represents the owner, Bob is a contributor and Carol and Edward are stakeholders. They specify their access control policies over the shared photo as follows:

$P = < Alic, OW, < (photo\ id\ , photo\ content), medium > , \{< travelling, GN >\}, \{\ \} >$

$P = < Bob, CB, < (photo\ id, photo\ content), high > , \{< David, UN >, \}, \{< family, RN >\} >$

$P = < Carol, ST, < (photo\ id, photo\ content), highest > , \{< family, RN >\}, \{< worker - friend, GN >\} >$

$P = < Edward, ST, < (photo\ id, photo\ content), low > , \{< family, RN >\}, \{< friend, RN >\} >$

Suppose that David (D), who is the accessor, has friend relationship with A, B, and C. He is applicable to access based on Alice's and Bob's policies. However, Carol's policy prevents David to view the shared item because he is member in her worker-friend group. David has not been mentioned in Edward's policy; as a result, Edward is not involved in the process of making collaborative decision. The positive and negative authorizations about David's access cause a conflict to decide either to allow or deny him. Thus, we apply the essential principles in our approach that are the weight of the accessor, the weights of the controllers, trust values and sensitivity level in order to resolve the conflicts. The permitted decisions value (*decision- permit*) is aggregated from Alice and Bob, who permit David to access. In contrast, the algorithm computes the value of denied decisions from Carol who does not allow David to view the photo. Suppose that trust values between David and each controller are inferred as shown in Figure 19.

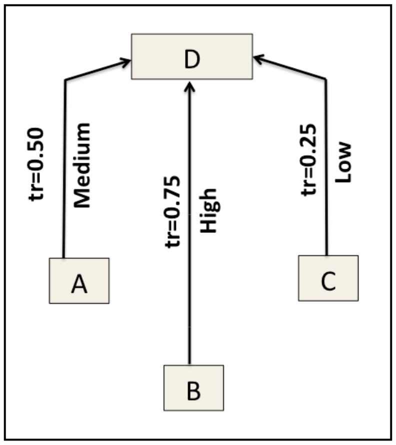


Figure 19: Inferring trust values.

For Alice's decision, owner's weight $\boldsymbol{w}(Alice)$ = 1 derived from Controllers' Weight scheme, $\boldsymbol{w}$*(David)* = 0.75 ,which is estimated based on the Accessors' Weight scheme, Alice trusts Dave .5 and sl = 0.50 ,which is derived from the $shared\ item$ of Alice's policy (p) . Consequently, the value of Alice's decision equals 0.188 based on the equation (2). On the other hand, Bob's policy leads us to contributor's weight $\boldsymbol{w}(Bob)$ = 0.50 is based on Controllers' Weight scheme that states when the contributor has a relationship or transitive relationship with the owner, her/his weight is 0.50, $\boldsymbol{w}$*(David)* = 1 and Bob trusts Dave 0.75 .Moreover, Bob assigns a high sensitivity level to the shared photo that is represented as sl = 0.75. Thus, Bob's decision is equal to 0.28; consequently, adding Alice and Bob's values, the value of the permitted decisions (*decision- permit*) based on equation (2) equals 0.468.

In contrast, to compute the value of the denied decisions that have arisen from Carol's policy (p), we infer $\boldsymbol{w}(Carol)$ = 1, ***w****(David)* = 0.75 because he is member in Carol's $worker - friend$ group and his privacy concern over the photo is in the highest sensitivity level that is equal to sl = 1. The trust level from Carol to David is low, valued at 0.25, and because this is a David is being denied by Carol, the trust is computed as 1 – 0.25. Subsequently, the value of denied decisions (*decision- deny*) equals 0.562 based on equation (3), which means the final result about David's access is to deny him to view the shared photo. Our strategy adapts varied factors in making collaborative decision rather than just a plain voting scheme; for instance, in this scenario we had two permitted votes and one deny vote but the final decision was denied the accessor to view shared item according to the PermittedandDeniedAccessors algorithm. Not that if letting the owner, Alice, taking full responsibility over who can access the shared photo, the decision would have been permitting only users who are members in her travelling group. Unfortunately, this limited decision of accessing shared item is what most of developed social networks (e.g. Facebook, Google+, MySpace, Twitter, etc.) offer.

4.3.2 AccessorSharing Algorithm

It is effective security practice to divide accessors into viewers and disseminators. The viewers are users who have permission to view the shared item. When a viewer requests to share the item with her/his friends, family members, classmates, etc. and is granted to disseminate the shared item, we call this viewer a disseminator. We introduce the AccessorSharing algorithm where the basic idea is using a trust value between associated controllers and accessors of the shared data as a threshold to decide whether the trust value is high enough to allow this. Algorithm 2 illustrates the entire procedure of accessor sharing.

Algorithm 2.AccessorSharing

```
input : ($item, P = \{< controller, TypeC, shared\ item, permitted\ accessor, denied\ accessor >\}$ ,
        ASP ={ < controller 1, tr-threshold > , . .  ,  < controller i, tr-threshold >},  final permitted
```

```
          accessor [ ] , TG ) // P set of p where each p ∈ P and is access control policy that is
          assigned by each associated controller for the data item in question , ASP set of asp that
          is accessor sharing policy where each controller identify her/his trust threshold value,
          final permitted accessor [ ] which is the output of PermittedandDeniedAccessors
          algorithm, $TG$ is trust graph where each user u ∈ U assigns a trust value to those who
          they have direct relationships with.

output : disseminators : int [ ] , not-disseminators : int [ ]

var
  tr                  : string   init null
  controllers-permit  : [ ]int   init null
  controllers-deny    : [ ]int   init null
  decision- permit    :double    init 0
  decision- deny      :double    init 0
  T                   : int      init 0
  F                   : int      init 0

begin
   for each  u ∈  final permitted accessor [ ]  do
     {
       for each controller c ∈ U do  // controllers who belong to ASP { }
         {
           tr ← $\boldsymbol{TG.infer}$(c, u) // using the FuzzyTrust algorithm to infer how much controller
                                     trusts user u from TG
           if  tr ≥ tr-threshold   // from ASP { } for controller c
              { controllers-permit [ ]  ← c
                T++}
           else
              { controllers-deny [ ]  ← c
                T++}
         }
       if   i = T  then  // checking if there are no conflicts where i is number of controllers (size of
                              ASP set )
           add u to disseminators [ ]
      else if  i= F then
           add u to not-disseminators [ ]
       else // conflict case
           for each controller c ∈ controllers-permit [ ]  do
                decision - permit += $\boldsymbol{w}(c) * \boldsymbol{sl}$(item)
           for each controller c ∈ controllers-deny [ ]  do
                decision - deny += $\boldsymbol{w}(c) * \boldsymbol{sl}$(item)
           if  decision - permit ≥  decision - deny then
                add u to disseminators [ ]
           else
                add u to not-disseminators [ ]
     }
  return disseminators [ ] and not-disseminators[ ]
```

This algorithm simply returns the set of accessors who are allowed to disseminate the shared item (*disseminators*) and a *not-disseminators* set that has accessors who are not allowed to disseminate. The algorithm takes set of policy (p) to determine the type controller and her/his sensitivity level of shared item. Moreover, Algorithm 2 takes as input the trust graph (TG), where a label is assigned to each edge to indicate the trust value of the relationship, to infer trust values (*tr*) between users. Also, a set of accessor sharing policy (*ASP*) that is defined by associated controllers is taken as input. In the first phase, the algorithm computes the trust value (*tr*) between each associated controller and accessor, who belongs to $final\ permitted\ accessor$ which is the result of the PermittedandDeniedAccessors algorithm by using the FuzzyTrust algorithm ($TG.infer(u)$). Then it compares the trust value (*tr*) with the sharing policy is specified by each controller (*tr-threshold*), which is part of the input of Algorithm 2. The *tr-threshold* in the sharing policy decides whether the trust value (*tr*) between the controller and the accessor is high enough for sharing or not. *tr-threshold* indicates how high the minimum trust value (*tr*) from the controller to the accessor should be to grant the accessor permission to share the item. If the trust value (*tr*) is equal to or higher than the required *tr-threshold*, the controller is added to the set of controllers (*controllers-permit*) that has all controllers who's sharing policy is achieved. Otherwise, controllers whose sharing policy requirements have not been satisfied are sent to *controllers-deny*. If the number of users in *controllers-deny* set equals the number of related ownerships, the accessor is not granted a permission to share the item with their friends, family members, classmates, etc. Indeed, the trust values between this denied accessor and controllers did not achieve the *tr-threshold* requirements; hence, the accessor is sent to the *not-disseminators*.

In the last case, when the value of trust (*tr*) from each controller to the accessor does not satisfy all sharing policies that are specified by controllers, who are associated with the shared item, a conflict arises among them to allow or refuse the sharing request. Consequently, we decide to solve conflicts by combining the controllers' weight and sensitivity levels that are derived from the shared item element policies and reflect the controllers' privacy concerns. We believe the relationship between controllers and shared items, which is represented by the controller's weight, has a significant impact to resolve conflicts. In addition, some controllers require high protection for shared items; consequently, combining sensitivity levels to make a decision is necessary to prevent inappropriate handling of data. To reach the final result, we compute a decision from controllers whose requirements degree of trust (*tr-threshold*) have not been achieved, as well as calculate a decision value from controllers whose assign trust level (*tr-threshold*) is satisfied by the value of (*tr)*. The values of permitted decisions and denied decisions are computed with following equations:

$$\text{decision - permit} \mathrel{+}= \boldsymbol{w}(c) * \boldsymbol{sl(}item) \qquad (5)$$

$$\text{decision - deny} \mathrel{+}= \boldsymbol{w}(c) * \boldsymbol{sl}(item) \qquad (6)$$

Finally, when the permitted decision value is greater than or equal to the denied decision value, the final result to disseminate the shared data item by the accessor is authorized. Otherwise, the accessor is refused to share the content by the AccessorSharing algorithm. To illustrate the algorithm's details, we introduce the following example.

Example 2.

Suppose there are four users, Alice (A), Bob (B), Clare(C) and Dave (D), who share the same photo where Alice has the photo in her profile, Bob and Dave are tagged in the photo that was initially was posted by Clare. Figure 12 shows a similar scenario where Alice is called the owner, Bob and Dave are stakeholders and Clare is called a contributor who posted the content in Alice’s profile. When Edward (E), who is one of viewers, desires to share this photo with his relationship list such as friends, family members, classmates, etc., authorization requirements from all linked controllers are considered by running the AccessorSharing algorithm. A, B, C and D have specified their accessor sharing policies (*tr-threshold*), and trust values between E and each controller are inferred as shown in Figure 20.

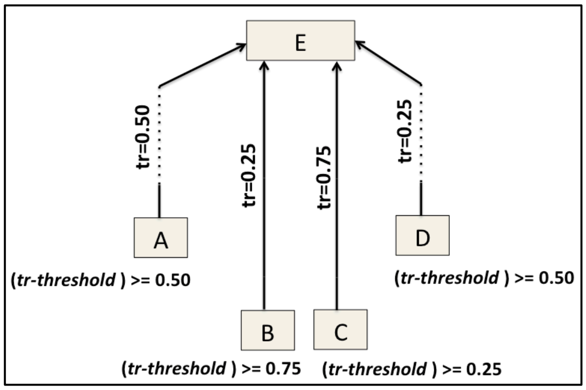


Figure 20: Accessor sharing policies and Inferring trust values.

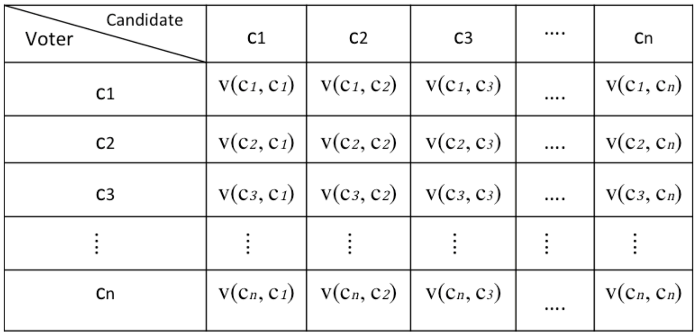
Also, Figure 20 depicts that the trust value from owner (A) to E satisfies A's trust requirement as well the contributor, Clare, trust requirement. However, stakeholders B and D refuse E to disseminate the shared photo because their policy requirements have not been satisfied. As a consequence, the decision for E's sharing request includes both permissions and prohibitions that lead to conflicts. For this example, suppose the photo has diverse sensitivity levels assigned by associated controllers A, B, C and D as follows *sl(photo(A))* = 0.25, *sl(photo(B))* = 0.75, *sl(photo(C))* = 0.50 and *sl(photo(D))* = 1. A and C's permissions are calculated by *decision – permit* =($w\ (A) *$ *sl(photo(A)))* + *(* $w(C) *$ *sl(photo(C)))* = 0.50 , where $w(A)$ and $w(C)$ are derived from the Controllers' Weight scheme. In order to resolve the conflict between controllers'policies about E sharing, Algorithm 2 computes a denied decision value from B and D based on the *decision – deny* equation that gives ($w(B) *$ *sl(photo(B)))* + *(* $w(D) *$ *sl(photo(D)))* = 1.75 , where B's and D's weights *w* are derived from principle 1. To acquire a final decision about E, the *decision – permit* is used as a threshold for decision making. In our example *decision – deny* is higher than *decision – permit,* thus E is denied to publish the photo on his profile. The collaborative decision for E's sharing request takes the privacy protection of highly sensitive data into account. Note this example also corresponds to the situation in Figure 14 and 15 where Edward reflects the contributor who desires to share and post the photo on his friend, family member, classmates, etc. personal page in OSN. Note that if the decision has been left up to the owner, Alice, the decision would have been permit Edward to share the photo with his relationships list. There is no consideration for the other parties' privacy requirements in most existing OSN privacy protection mechanisms.

Furthermore, they do not support trust notion and sensitivity level of data, which play significant role to measure how disclosure item can affect online users.

4.3.3 ControllerSharing Algorithm

Shared data item dissemination comes in two varieties, both useful to increase the communication among users in OSNs. The first sharing model was previously introduced as accessor sharing. In this section, we are going to present a second way to disseminate a shared data item, and an algorithm which describes how possible conflicts between associated controllers should be solved. According to case one in the sharing scenarios, shown previously in Figures 8 and 9, a sharing request comes from one of the controllers, who is related to the shared item. When a controller desires to share the item with her/his friends, family members, classmates, etc., the item will be in turn posted in a new user’s profile; this new user could be unauthorized to access the shared item. Therefore, the controller sharing request has a high potential to leak data. We have formalized the ControllerSharing algorithm to enable controllers to regulate their privacy and protect items from being used against them in some way.

Before we express algorithms in pseudo code, we need to elucidate some basic ideas behind the ControllerSharing algorithm. The first idea is a boolean voting matrix (0's and 1's) of size $|C| \times |C|$, where C is the set of controllers, which contains all sharing policies of a particular shared item where rows refer to controllers as voters and each column denotes controller as candidates and her/his receiving votes. In the voting matrix, all controllers can vote for or against all other associated controllers to share the content, which enables them to express their opinions comprehensively.



Candidate Voter	c1	c2	c3	....	cn
c1	$v(c_1, c_1)$	$v(c_1, c_2)$	$v(c_1, c_3)$	....	$v(c_1, c_n)$
c2	$v(c_2, c_1)$	$v(c_2, c_2)$	$v(c_2, c_3)$	....	$v(c_2, c_n)$
c3	$v(c_3, c_1)$	$v(c_3, c_2)$	$v(c_3, c_3)$	....	$v(c_3, c_n)$
⋮	⋮	⋮	⋮	⋮	⋮
cn	$v(c_n, c_1)$	$v(c_n, c_2)$	$v(c_n, c_3)$	....	$v(c_n, c_n)$

Figure 21: The Voting Matrix of Controllers Sharing.

A cell in the voting matrix with row $c1$ and column $c2$ is filled with $v(c1, c2)$, which is the value of the vote that $c1$ confers on c2. If $v(c1, c2) = 0$ that means $c1$ prevents $c2$ to disseminate the shared item; in contrast, $v(c1, c2) =1$ indicates that $c1$ allows $c2$ to share the item. However, $v(ci, ci)=1$ for all i.

In the ControllerSharing algorithm, the trust value (*trv*) of the voter is calculated differently than in prior algorithms where $TG.infer\ (controller \rightarrow accessor)$ denotes to how much the controller at the tail of the edge trusts the accessor at the head of the edge, which may not be directly connected in the trust graph. However, trv_{voter} refers to the trust value of the voter that is computed based on the following equation:

$$trv_{voter} = \sum_{j=1}^{|C|} \boldsymbol{TG.infer}\,(j \longrightarrow voter) \qquad (7)$$

where j is candidate and $TG.infer\ (j \longrightarrow voter)$ calculates the trust value from each candidate, who belongs to the associated controllers set, to the voter. trv_{voter} indicates how much related controllers trust this controller ($voter$). Each individual $TG.infer$ $(j \longrightarrow voter)$ value in equation (7) is calculated by the FuzzyTrust algorithm. Consequently, the trust inference algorithm computes the trust value between diverse sources to a particular voter. The ControllerSharing algorithm is detailed in Algorithm 3.

Algorithm 3. ControllerSharing Algorithm

```
input : $(item, P = \{< controller, TypeC, shared\ item, permitted\ accessor, denied\ accessor >\},$
        voting-matrix (voter, candidate), TRV= $\{trv_0 , . . . ,\ trv_i\})$// P set of p where each $p \in P$ and is
        access control policy that is assigned by each associated controller for the data item in
        question, voting-matrix giving the controller sharing policy from each controller, TRV set of trv
        that is the trust value of each controller (voter) has been computed based on equation (7).

output : $permitted\ controllers$ : int  [ ] , $denied\ controllers$ :int [ ]

var
  vote- permit    : int    init 0
  vote- deny      : int    init 0

begin
   for each candidate c (in column) do // running in voting-matrix
     {
       for (i=0 ; i<= |C| (#voter in row) ; i++)
           if  voting-matrix ( $i$, c) =1 then  // checking if there are no conflicts
               add candidate c to $permitted\ controllers$ [ ]
        else // conflict case
            for each voter $i$ do
                {
                    if voting-matrix (voter $i$, candidate c) =1 then
                        vote- permit += $\boldsymbol{trv}_{(\boldsymbol{voter}\ i)} * \boldsymbol{w}(voter\ i)$ // $trv_{(\boldsymbol{voter}\ i)}$ from $TRV\{\ \}$
                    else
                        vote- deny += $\boldsymbol{trv}_{(\boldsymbol{voter}\ i)} * \boldsymbol{w}(voter\ i)$
                }
    if vote- permit $\geq$ vote- deny then
        add candidate c to $permitted\ controllers$ [ ]
    else
        add candidate c to $denied\ controllers$ [ ]
  }
return $permitted\ controllers$ [ ] $and\ denied\ controllers$[ ]
```

Figure 8 and 9 demonstrated the situation where one of controllers desires to disseminate a shared item; before it is posted in a new space, we investigate if the sharing requester is allowed to share or not by checking the ControllerSharing algorithm result. The ControllerSharing algorithm uses a voting matrix and set of policies (P) to determine the controller's weight as input. Additionally, Algorithm 3 receives the set of the trust values of the voter (trv) that represents how much associated controllers trust this controller ($voter$) based on equation (7). The ControllerSharing algorithm produces two sets of controllers, who, based on conflict resolution strategy in the algorithm, could be permitted ($permitted\ controllers$) to share or denied ($denied\ controllers$), as output.

In the ControllerSharing algorithm, for every candidate appearing in the voting matrix, the function *getvoting-matrix* translates candidate votes into a *voting-value*. If the candidate received a vote=1 from all voters then add this candidate to the $permitted\ controllers$ set which a list of controllers who are allowed to disseminate the shared item. On the other hand, some conflicts might occur where we have both permitted and banned votes for a particular candidate to share. Hence, we use equations (8) and (9) in our conflict resolution strategy to manage conflicts about a candidate,

$$\text{vote- permit} \mathrel{+}= \boldsymbol{trv}_{(\boldsymbol{voter}\ i)} * \boldsymbol{w}(voter\ i) \tag{8}$$

$$\text{vote- deny} \mathrel{+}= \boldsymbol{trv}_{(\boldsymbol{voter}\ i)} * \boldsymbol{w}(voter\ i) \tag{9}$$

where $trv_{(\boldsymbol{voter}\ i)}$ is the trust value of the voter that is computed by the FuzzyTrust algorithm based on equation (7) and $w(voter\ i)$ is the weight of that voter that is calculated by the controller weight principle. The trust value and weight of the voter are considered as essential elements to solve conflicts between voters' authorizations.

After we compute both permitted votes and prohibited votes, the final decision about a candidate can be made as:

$$\begin{array}{l} \text{vote- permit} \geq \text{vote- deny} = \text{selected candidate} \\ \text{vote- permit} < \text{vote- deny} = \text{unselected candidate} \end{array} \qquad (10)$$

The algorithm makes a final decision based on values of voters that are in *vote- permit* and *vote- deny*. The candidate(s) with a higher *vote- permit* value will be allowed to disseminate the shared item with her/his friends, family members, classmates, etc. Nevertheless, when voters who vote against the candidate produced value of denied decision (*vote- deny*) that is higher than value of the permitted decision (*vote- permit*), the candidate is denied to share.

As we discussed in the previous algorithms, the last result of our approach moves toward the permitted decision to maintain the social value of data sharing in OSNs. Finally, the algorithm returns a set of controllers who are allowed to disseminate shared items and a set of banned controllers. The following example will illustrate ControllerSharing algorithm precisely.

Example 3.

Suppose there are three controllers Alice (A), Bob (B), and Clare(C) who specify access control policies (*P*) to control who can view a shared picture. Alice owns the shared picture which originally was posted in her space by Clare who tagged Bob in this picture. Consequently, Alice is the owner, Clare is considered to be the contributor and Bob, who is tagged in picture, is called a stakeholder. All associated controllers are required to specify a controller sharing policy. Then, a voting matrix is made from Alice's, Bob's and Clare's policies, as shown below in Figure 22.

Candidate / Voter	A (owner)	B (stakeholder)	C (contributor)
A (owner)	1	0	1
B (stakeholder)	1	1	1
C (contributor)	0	0	1

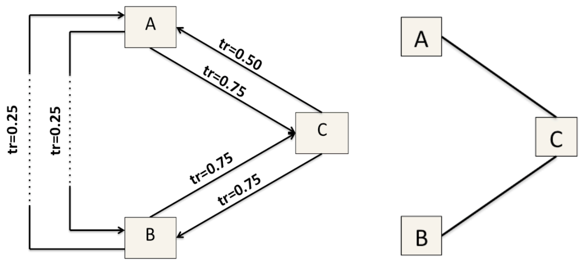


Figure 22: The voting matrix (left), a directed trust graph, and an undirected relationship graph (right).

Bob wants to share the picture with his friend; therefore we check if he is able to share the photo based on the ControllerSharing algorithm. Suppose the trust values between controllers are inferred as presented in the directed trust graph in the middle of Figure 22. According to the voting matrix, Bob has two against votes and one for vote, then his situation has conflicting votes. Algorithm 3 computes *vote- permit* that contains all voters who vote for Bob to share. On the other hand, the total of votes against Bob produces *vote- deny* value based on equation (9). *vote- permit* equals 1 based on equation (8) where $trv_{(\boldsymbol{B})}$=1 that is calculated by equation (7) as follows : $\boldsymbol{TG.infer}\ (A \longrightarrow B) + \boldsymbol{TG.infer}\ (C \longrightarrow B)$,where we suppose the trust values are inferred as shown in the directed trust graph Figure 22 ; subsequently, $trv_{(\boldsymbol{B})}$= $\boldsymbol{tr}\ (A \longrightarrow B) + \boldsymbol{tr}\ (C \longrightarrow B)$ = 0.25 + 0.75 =1. $w(B)$ reflects stakeholder's weight = 1 based on principle 1, hence *vote- permit* =1. *vote-deny* = ($(trv_{(\boldsymbol{A})} * w(A)) + (trv_{(\boldsymbol{C})} * w(C))$), on the basis of equation (7) and according to inferred trust values in Figure 22 $trv_{(\boldsymbol{A})}$ = 0.75 and $trv_{(\boldsymbol{C})}$ =1.50. Where $w(A)$ reflects the owner's weight =1 and $w(C)$ reflects a contributor 's weight = 0.50 which are based on the Controllers' Weight Scheme. As a result, vote- *deny* =1.5, so the final decision denies Bob to disseminate the picture. Note that the decision for regulating the sharing policy to the shared item still rests solely on the owner of shared item, Alice, in most OSN platforms (e.g. Facebook, Google+, MySpace, Twitter, etc.). As such, in these OSN sites the decision would have been based on Alice privacy without dealing with the privacy concerns of other users, Clare and Bob, who appear in the photo.

4.4 Summary

In summary, existing OSNs support privacy decisions as individual processes regardless of other associated users' privacy concerns. Basically, users are sharing their life actions and events with their social network. Consequently, most data in OSNs have multiple connected users who desire to manage their data. Sharing patterns are showing the risk that is posed by lacking a collaborative privacy management framework in OSNs. In addressing the limited access control in current OSNs, we have proposed a methodology for collaborative access control in OSNs. First, by abstract representation of OSNs, we identified the key elements of OSN to build and characterize our model and then represent individual access control policies where each controller, who belongs to a set of collaborative controllers for particular shared items can specify her/his privacy policy over the shared item. After local specifications, we have discussed the principles of our conflict resolution strategy to solve the conflicts in contradictory privacy requirements that are caused by different controllers' privacy preferences. We have discussed and investigated how controllers' weight, accessors' weight and inferring fuzzy trust can help to resolve the conflicts and to reach a collective privacy decision. Furthermore, we have explored how a collaborative access control model based on these principles with our algorithms can be achieved. Our PermittedandDeniedAccessors algorithm makes a concerted attempt to reach a collaborative decision over who can access the shared item and who cannot. Lastly, we have introduced our AccessorSharing and ControllerSharing algorithms to collaboratively regulate who can disseminate the shared item.

Chapter 5

5 Implementation and Evaluation

This chapter describes the implementation phase of our Permitted and Denied Accessors algorithm to demonstrate the efficacy of the approach.Our prototype application is begun by generating multiple controllers' scenarios, where more than one user should be involved in the process of specify the access control policies. We implemented our prototype as a small programming project using Java and the MySQL Database. In fact, the Java programming language is very appropriate because it offers an easy mechanism called a package that builds a namespace for each component of the prototype.

We briefly review the notion of the PermittedandDeniedAccessors algorithm in order to provide the idea of how we implemented the collaborative decision. The algorithm generates two sets of accessors as output. The first set contains accessors who are permitted to view the shared data; in contrast, the second set includes those who are denied. The algorithm receives a policy (*p*) from each controller of the targeted item as input. To review the algorithm, two steps are performed to get the final sets of accessors. In the first step, we determine the final decision concerning a certain user; if all associated controllers agree to permit or deny her/him, the algorithm can yield the final decision without moving to conflict resolution phase.

Indeed, conflicts might always exist when taking into account multiparty control over the shared item because co-controllers of shared items often have different privacy concerns over the data. In the second step of Algorithm 1, we attempt to reach the final decision by resolving the privacy policy conflicts. The combined use of our principles (Controllers' Weight Scheme, Accessors' Weight Scheme, Inferring Fuzzy Trust) has been implemented in the second step of the PermittedandDeniedAccessors algorithm to resolve conflicts. These principles are adopted as a convenient way to give the ability to each controller associated with a shared item to affect the final decision for viewing a shared item. For a user who receives conflicted policies about her/his access, Algorithm1 evaluates this final decision by separately aggregating the value of permitted decisions and denied decisions. Finally, the aggregated permitted decision value is utilized as a threshold for making the final decision. For more detail about this algorithm, see Chapter 4. Algorithms 2 and 3 are not currently implemented; their evaluation is part of our future work.

We organize this chapter as follows. Sect.1 we present our dataset, which is used for running our algorithm and creating multiple controllers' scenarios. In Sect. 2 we show all sharing cases occurring in OSNs, where different controllers may have diverse access control and privacy policies for a single content. The final section provides a complete multiple-controllers' scenario and compares the final decision that is produced from our algorithm with what the current Facebook decision would be.

5.1 Dataset

In this section we describe our test dataset for the PermittedandDeniedAccessors algorithm. It involves what we hope is a realistic environment and simulates our relationships network and trust network. Also, it is important to generate multiple controllers' scenarios efficiently. Our dataset is created in MySQL as 9 tables in a structure that fundamentally simulates the Facebook application. However, we additionally add some features in our database structure such as relationship types and a trust relationship to test our prototype in a suitable environment. Our OSNWS database has 30 users who are connected to each other by 363 diverse relationships. We have 4 relationship types in our dataset: friendship, family, classmate and colleague. Furthermore, in the database there are 120 groups where each user has 4 groups. Additionally, each user randomly belongs as a member to several groups, which are owned by different users, hence the number of members in all groups, is 279. To measure the PermittedandDeniedAccessors algorithm's performance in terms of efficacy, we design 168 shared contents that are uploaded by the users in the database and are disseminated among users. Also, to have shared content, we tag different users in each shared item. Thus the total number of tagged users is 364.

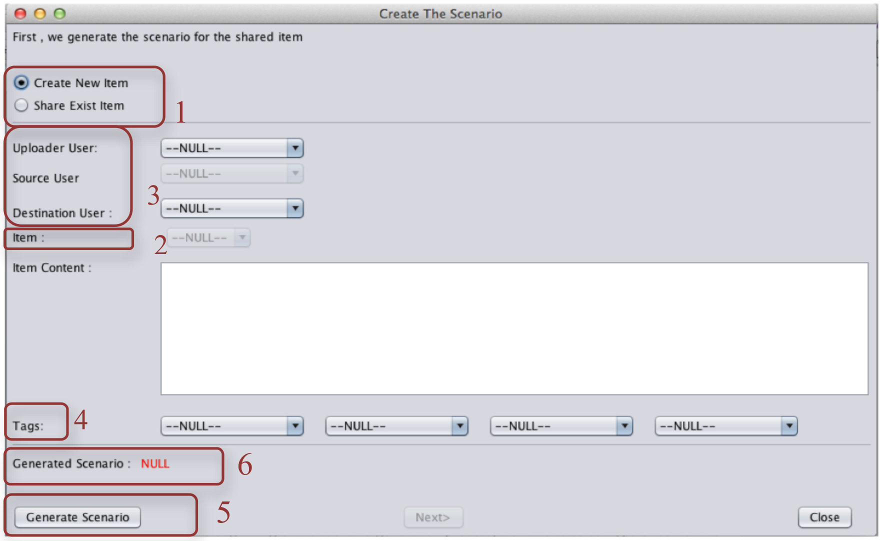
In our prototype we can simulate scenarios which occur by creating new shared items or sharing items which already exist in the database, as shown in Figure 23. In conclusion, we attempt to design our dataset to be a realistic environment that the prototype can efficiently implement.

5.2 Multiple Controllers’ Scenarios and Policy Specification

In this section, we are going to generate all scenarios where more than one user may be involved in the access control process. Actually, multiple controllers’ scenarios were introduced and analyzed in Chapter 3. However, we generate those scenarios as the first phase of our prototype because they lead us to determine all users who have the right to participate in the process of making a collective access control policy. Moreover, simulating these scenarios is a critical part to successfully testing the PermittedandDeniedAccessors algorithm, which reaches a collective decision over who can access the shared content. Furthermore, we present the simulation of the collaborative policies for authorization administration.

5.2.1 Multiple Controllers’ Scenarios

According to the Content Sharing Section 3.3 in Chapter 3, we organize our simulations here. First, we explain the interface for creating a scenario as shown below in Figure 23.



Create The Scenario

First , we generate the scenario for the shared item

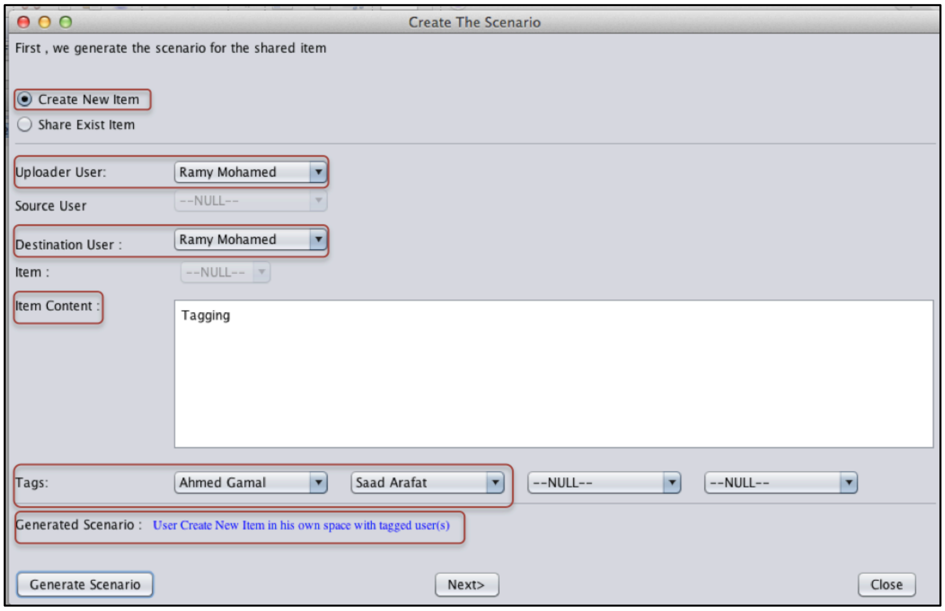
● Create New Item
○ Share Exist Item

Uploader User: --NULL--
Source User --NULL--
Destination User : --NULL--
Item : --NULL--
Item Content :
Tags: --NULL-- --NULL-- --NULL-- --NULL--
Generated Scenario : NULL

Figure 23: Interface of creating scenarios.

At the beginning of the scenario generation process, we have to select type of content as shown in box1. If the content already exists, the list of existing items will be active in box 2. Box 3 shows the user who uploads the item and a destination user who owns the profile where uploader posts the item. The source user will be active when we choose to share the existing item, so all items that belong to the source user will appear in the item drop list shown in box 2. Furthermore, when the uploader desires to tag other users in a content, our prototype allows him/her to tag four users as shown in box 4. Thus, using the button shown in box 5, we can create a scenario based on previously determined information. Finally, the status of the generated scenario will be changed (box 6), and then we specify our collaborative policies by moving to next interface.

The first case is the Tagging scenario that consists of an uploader who is the owner and a few stakeholders who are tagged users, as previously analyzed in Figure 5.



First , we generate the scenario for the shared item

● Create New Item
○ Share Exist Item

Uploader User: Ramy Mohamed

Source User --NULL--

Destination User : Ramy Mohamed

Item : --NULL--

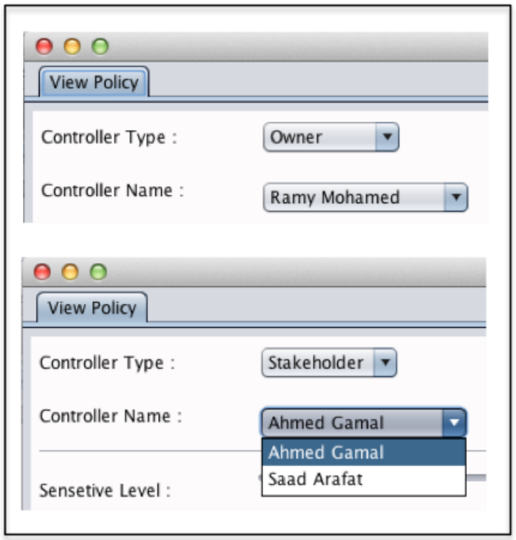
Item Content : Tagging

Tags: Ahmed Gamal | Saad Arafat | --NULL-- | --NULL--

Generated Scenario : User Create New Item in his own space with tagged user(s)

Figure 24: Tagging scenario

Figure 24 presents a tagging scenario where Ramy uploads a new item in his profile and tags two users. The scenario generated has two types of controllers, as shown in the policy interface below.



Controller Type : Owner

Controller Name : Ramy Mohamed

Controller Type : Stakeholder

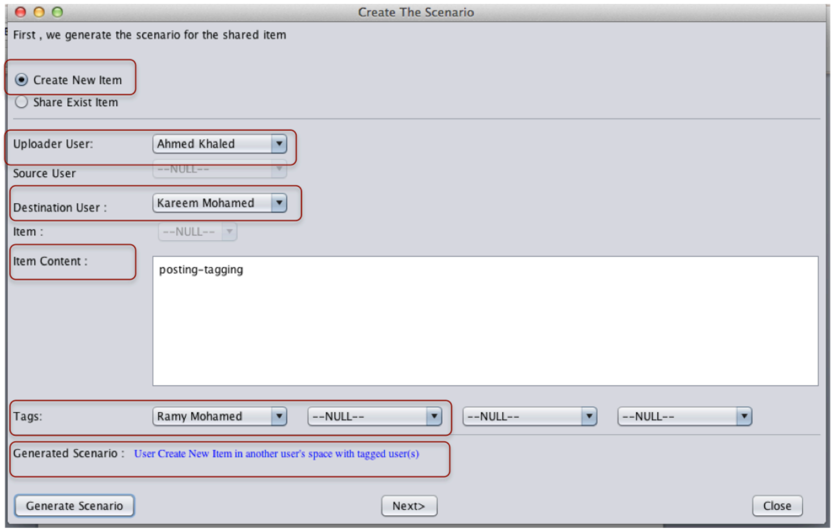
Controller Name : Ahmed Gamal

Sensetive Level :

Figure 25: Co-controllers in Tagging scenario.

Here we consider Ramy as owner and tagged users Ahmed and Saad as stakeholders. To produce collaborative policies over who can view the shared item, we aggregate policies from owner and stakeholders.

Another sharing tool, for posting, (the second scenario), is shown in the following Figure 26.



First , we generate the scenario for the shared item

● Create New Item
○ Share Exist Item

Uploader User: Ahmed Khaled

Source User --NULL--

Destination User : Kareem Mohamed

Item : --NULL--

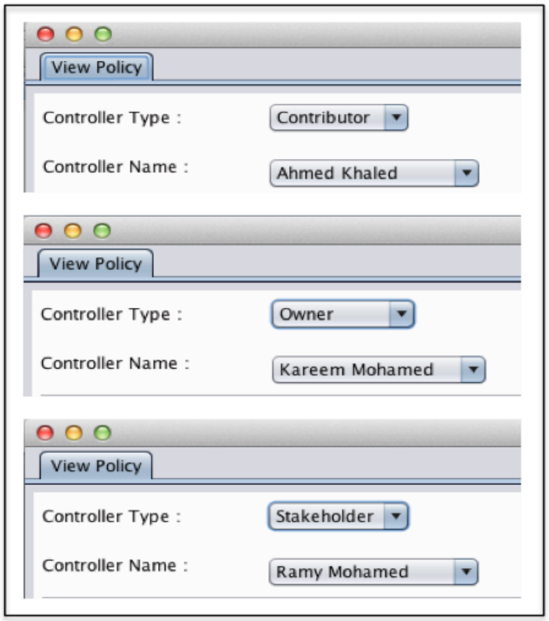
Item Content : posting-tagging

Tags: Ramy Mohamed --NULL-- --NULL-- --NULL--

Generated Scenario : User Create New Item in another user's space with tagged user(s)

Figure 26: Posting-Tagging scenario.

The uploader posts an item in Kareem's profile and tags Ramy; thus, we have Ahmed as contributor, Yasser as owner and Ramy as stakeholder. In Chapter 3, we investigated the Posting -Tagging scenario and Figure 6 shows this case. We let all associated controllers participate in the process of making the collaborative decision over who can access a shared item, as shown in the following figure.



Controller Type : Contributor

Controller Name : Ahmed Khaled

Controller Type : Owner

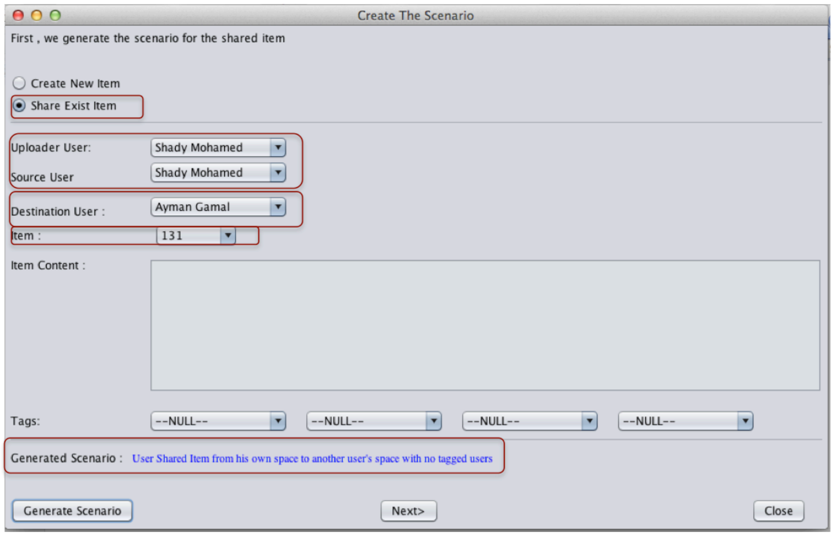
Controller Name : Kareem Mohamed

Controller Type : Stakeholder

Controller Name : Ramy Mohamed

Figure 27: Co-controllers in Posting-Tagging scenario.

Third is the sharing tool that is offered by most OSN applications to encourage their online users to share and distribute pictures, events or activates among their friends, family members or even co-workers in various ways. When users share their contents with others in their social network or share others users' contents, a simple sharing scenario occurs which was previously explained in Chapter 3, Figures 7 and 10. We simulate this scenario as follows, Shady is a user who has the item in his profile and he is the uploader of this item. Then, he shares it with Ayman who owns the destination profile, so we consider him the owner of the shared item. On the other hand, Shady becomes the originator of the shared item. Figure 28 presents the simple sharing scenario creation; afterward, Figure 29 shows the multiple controllers who collaboratively set the privacy polices for the shared item. According to our principles the originator's policy has less effect on the final access control policies compared to the new owner.



First , we generate the scenario for the shared item

○ Create New Item
● Share Exist Item

Uploader User: Shady Mohamed

Source User Shady Mohamed

Destination User : Ayman Gamal

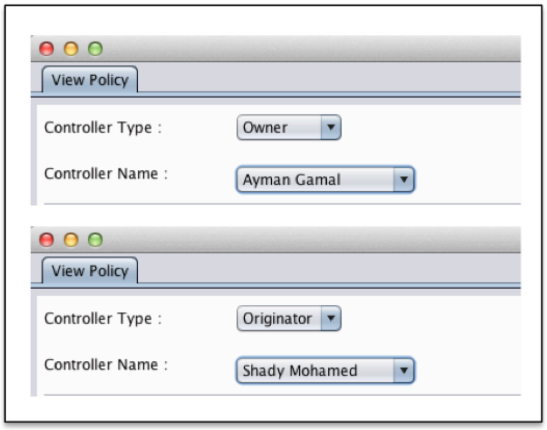
Item : 131

Item Content :

Tags: --NULL-- --NULL-- --NULL-- --NULL--

Generated Scenario : User Shared Item from his own space to another user's space with no tagged users

Figure 28: Simple Sharing scenario.



Controller Type : Owner

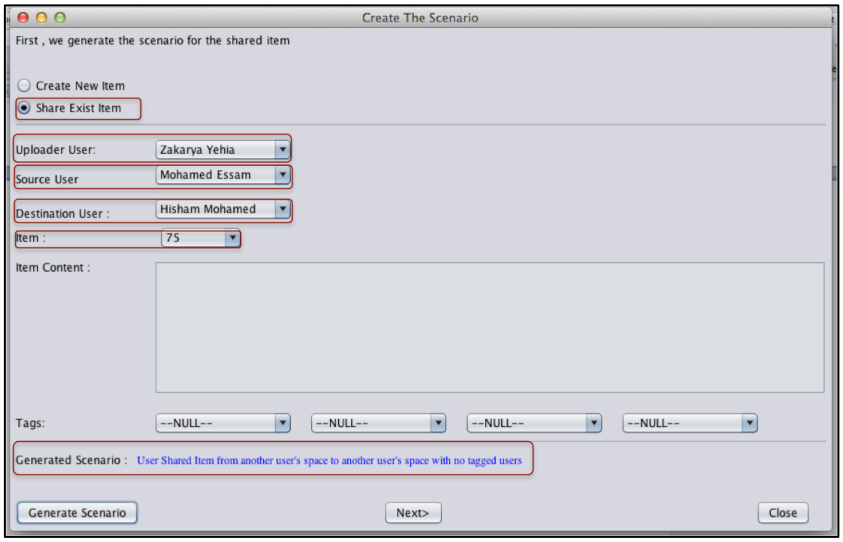
Controller Name : Ayman Gamal

Controller Type : Originator

Controller Name : Shady Mohamed

Figure 29: Co-controllers in Simple Sharing scenario.

By the sharing tool, diverse scenarios can arise such as a user sharing another user's content and posting it in someone else's space, which is more complicated than the previous scenario. When an intermediate user is involved in a sharing scenario, the process of making a collective decision has three types of controllers as clarified previously in Chapter 3 and analyzed in Figure 13. In the next figure we simulate the scenario where a user shares other user's content and posts it in someone else's space.



First , we generate the scenario for the shared item

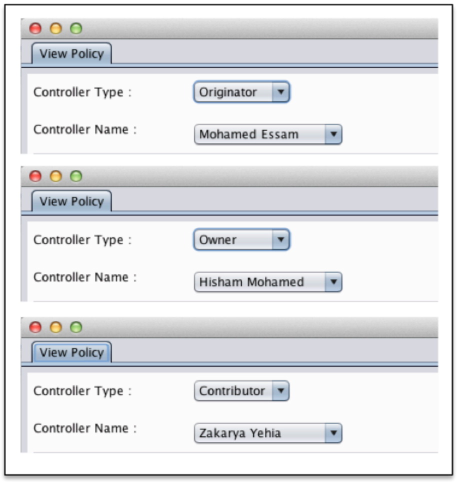
○ Create New Item
● Share Exist Item

Uploader User: Zakarya Yehia
Source User Mohamed Essam
Destination User : Hisham Mohamed
Item : 75
Item Content :
Tags: --NULL-- --NULL-- --NULL-- --NULL--

Generated Scenario : User Shared Item from another user's space to another user's space with no tagged users

Figure 30: Simple sharing of another users' item and posting it in someone else's space scenario.

Zakarya desires to share Mohamed's item 75 with Hisham. Thus, item 75 will be in turn posted in Hisham's space, who thus becomes the new owner of Mohamed's item. We call Mohamed the originator and Zakarya the contributor who reposts the item. Afterward, each associated controller has the right to regulate her/his privacy policies concerning the shared item 75. Figure 31 shows how our model considers all linked ownerships in the process of making a collaborative decision over who can access item 75 in Hisham's profile.



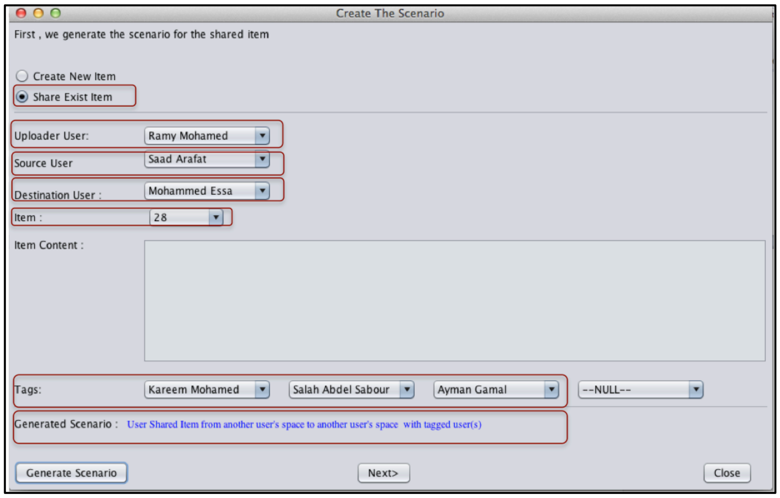
Controller Type : Originator
Controller Name : Mohamed Essam

Controller Type : Owner
Controller Name : Hisham Mohamed

Controller Type : Contributor
Controller Name : Zakarya Yehia

Figure 31: Co-controllers in a sharing scenario created by an intermediate user.

The final simulated scenario contains all types of ownership in our model. Suppose Saad's originates item 28 that Ramy desires to share with his friend Mohammed and Saad tags Kareem, Salah and Ayman in it. Hence, our model gives all involved users the right to participate in the access control process under different ownerships. When item 28 is reposted in Mohammed's space, Ramy becomes a contributor, Mohammed is the owner, and Saad is the originator. The collaborative access control polices of the shared item 28 in Mohammed's space, which are given by the result of the PermittedandDeniedAccessors algorithm, are regulated by these controllers who are related to the shared item by varied relationships. This case is represented in Figures 32 and 33.



Create The Scenario

First , we generate the scenario for the shared item

○ Create New Item
● Share Exist Item

Uploader User: Ramy Mohamed

Source User Saad Arafat

Destination User : Mohammed Essa

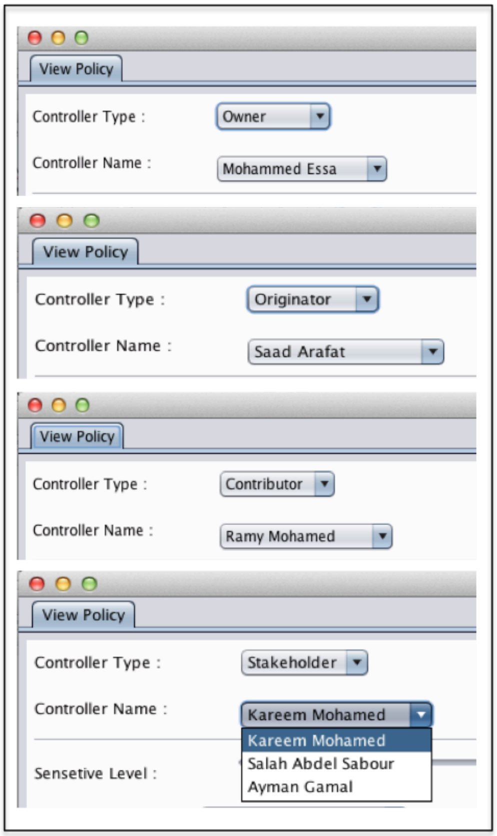
Item : 28

Item Content :

Tags: Kareem Mohamed, Salah Abdel Sabour, Ayman Gamal, --NULL--

Generated Scenario : User Shared Item from another user's space to another user's space with tagged user(s)

Figure 32: The final simulated scenario.



Controller Type : Owner

Controller Name : Mohammed Essa

Controller Type : Originator

Controller Name : Saad Arafat

Controller Type : Contributor

Controller Name : Ramy Mohamed

Controller Type : Stakeholder

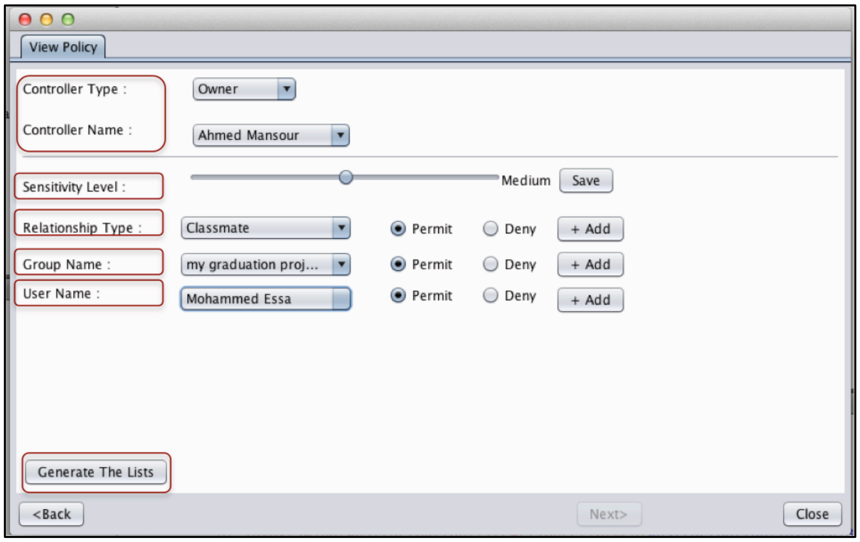
Controller Name : Kareem Mohamed

Sensetive Level :

Figure 33: Co-controllers of final simulated scenario.

5.2.2 Privacy policy specification

Associated controllers are required to feed input into the prototype, as complete privacy policies are necessary for the algorithms. We presented our collaborative policy specification scheme for access control and authorization administration in Chapter 4. Our policy privacy specification scheme includes two parts, first is the accessor specification which was defined in Definition 5 that supports both positive and negative authorizations. Controllers are able to regulate accessors, with whom they want to share their data, and unauthorized users. The second part is the data specification where a controller can determine the level of sensitivity of an item based on how much its disclosure would harm her/him. The interface for policy specification in our prototype is presented in Figure 34.



Controller Type : Owner

Controller Name : Ahmed Mansour

Sensitivity Level : Medium

Relationship Type : Classmate ● Permit ○ Deny

Group Name : my graduation proj... ● Permit ○ Deny

User Name : Mohammed Essa ● Permit ○ Deny

Figure 34: Privacy policy specification interface.

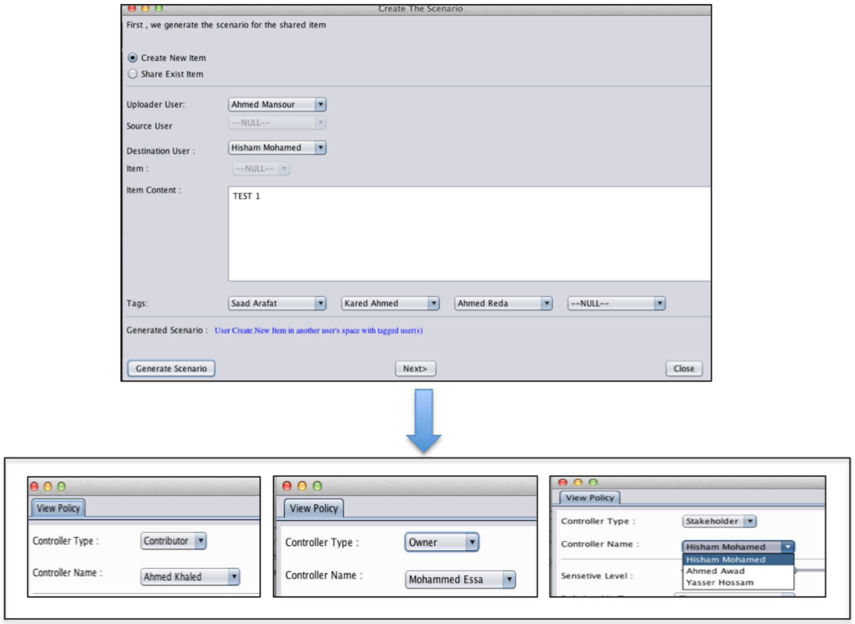
Through this interface, we obtain the set of individual AC policies (*P*). Each associated controller first determines the sensitivity level of the shared item and then specifies the authorized users and unauthorized users by three parameters that are user names, group names and relationship types. A drop list of relationship types, in the previous figure, shows all relationships Ahmed has in his social network. Likewise, the group list presents all groups that the owner has in his profile. Also, a controller can specifically permit or deny particular user from the user name list. After each controller regulates her/his privacy policy, our algorithm runs to generate a collaborative decision concerning which users can access the shared item and who cannot. Consequently we will receive two lists of accessors as output.

5.3 Experiments and Analysis of Results

In this section, we study a prototype of our collaborative access control model for accessing shared item situation (PermittedandDeniedAccessors algorithm). The purpose of our experimental study is to assess the effectiveness and usefulness of our model in terms of making a collaborative decision that balances between privacy requirements and sociability on OSNs. The results of our approach are compared to Facebook in certain scenarios where multiple users have the right to participate in the process of making final decision of shared items. These comparisons show the lack of a joint administration policy in Facebook and the feasibility of our collaborative access control model. Facebook’s results are computed based on its current privacy settings that take into account solely the owner’s privacy setting. Positive policy in Facebook can be customized with users names, group names and friendships. However, Facebook’s negative authorization can be specified by users names and group names only. Accordingly, in our prototype we consider the owner’s permitted and denied policies for Facebook results. The experimental results are obtained from our dataset where relationships and trust values between 30 users are randomly generated. In what follows, we evaluate the performance of our approach for 2 cases: sharing a new item and sharing a shared item, comparing to Facebook in similar scenarios.

5.3.1 Sharing a new item Experiment

Here we present the results acquired from the scenario of sharing a new item. The experiment runs as a posting-tagging scenario where the item has an owner, multiple stakeholders and a contributor, explained, previously in Figure 6. The scenario is generated and each associated controller regulates her/his privacy policy, as illustrated in Figure 35.



First , we generate the scenario for the shared item

● Create New Item
○ Share Exist Item

Uploader User: Ahmed Mansour
Source User --NULL--
Destination User : Hisham Mohamed
Item : --NULL--
Item Content : TEST 1
Tags: Saad Arafat | Kared Ahmed | Ahmed Reda | --NULL--
Generated Scenario : User Create New Item in another user's space with tagged user(s)

Controller Type : Contributor
Controller Name : Ahmed Khaled

Controller Type : Owner
Controller Name : Mohammed Essa

Controller Type : Stakeholder
Controller Name : Hisham Mohamed

Figure 35: The execution flow of sharing new item.

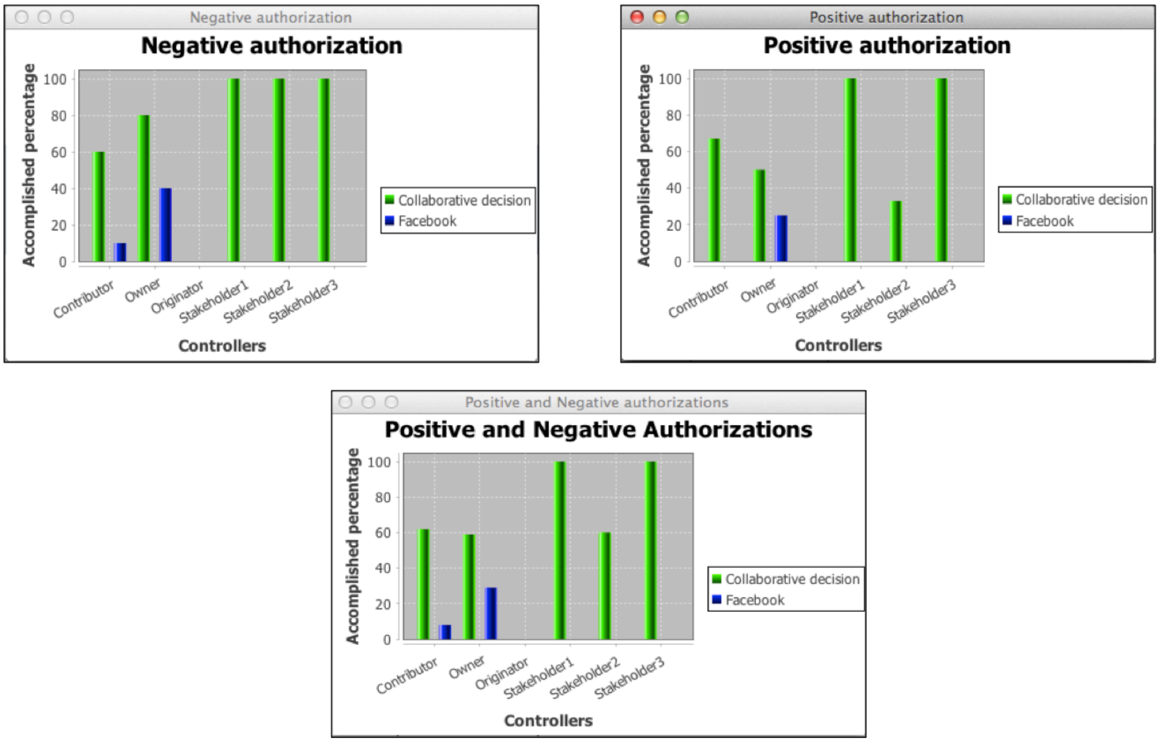


Figure 36: A comparison of our collaborative model and Facebook achievements for co-controllers privacy polices in a shared item scenario.

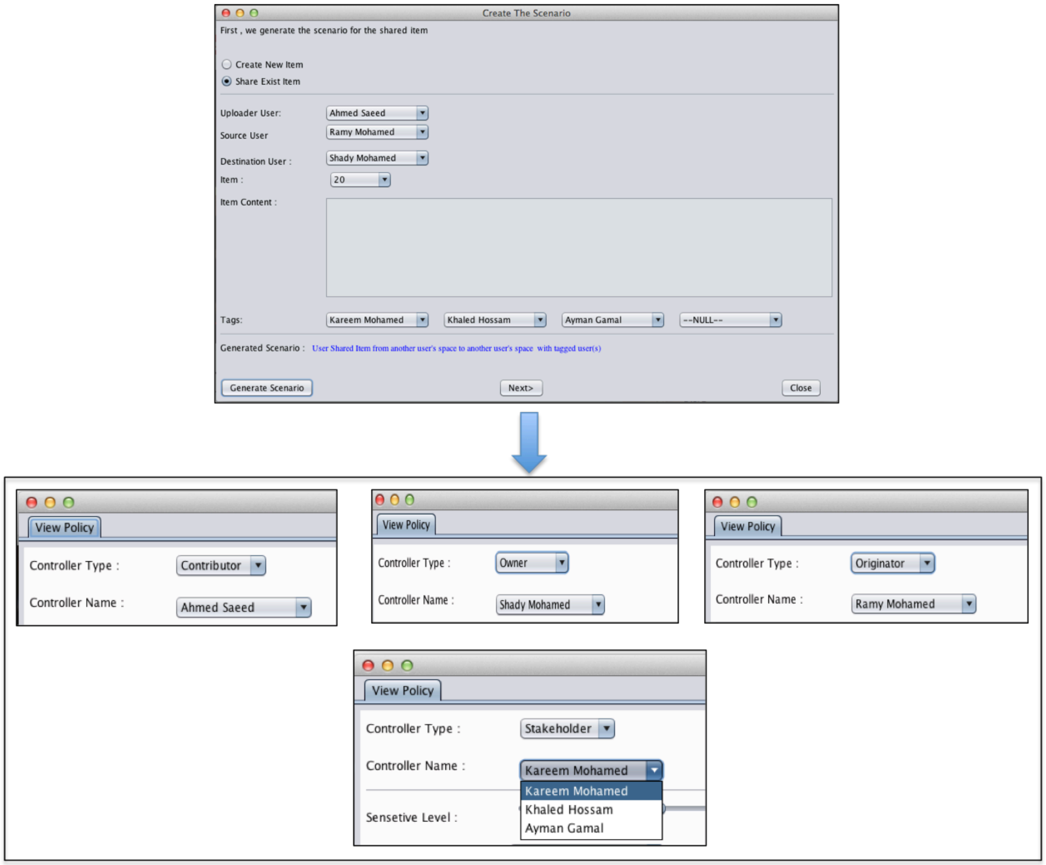
By running the PermittedandDeniedAccessors algorithm we obtain a collaborative decision concerning permitted accessors and denied accessors. The results from running Algorithm 1 and Facebook's method are displayed in Figure 36. The bars represent the percentage of the privacy policies achieved that are required from the enrolled co-controllers in our approach and what would be achieved by Facebook. The display allows us to compare the policy achievement of the associated controllers for our collaborative decision and for Facebook.

The first graph describes the accomplished amount of negative privacy policies of each involved controller for the two techniques, as a percentage. These quantities are calculated based on how much our model satisfies each co-controllers' negative privacy requirements; on the other hand, the blue bars present the amount of negative policy of all associated controllers achieved based on Facebook's setting. Although in the owner case both offer reasonable achievement, our model meets most of the stakeholders' negative policies. In fact, our dataset is a connected graph where there is a path between any two users. Thus, there is a high probability that an owner and another co-controller have denied accessors in common. We note that some of the contributor’s negative policy is accomplished by Facebook.

The second chart represents the percentages of the positive privacy requirement of each co-manager achieved by our model and by Facebook. As shown in the positive authorization graph, our approach attempted to resolve the conflicts between the owner, contributor and the stakeholder 2 in permitted accessors based on the conflict resolution strategy. However, accessors, who are authorized by stakeholders 1 and 3, do not affect other co-controllers’ privacy requirements. Finally, both permitted and denied policies are combined to measure the percent of total accomplishment of each co-controller’s privacy policy that is achieved by our collaborative model and Facebook, as presented in the positive and negative authorizations graph.

5.3.2 Sharing shared item Experiment

To demonstrate the usability of the approach in terms of the need of devising a collaborative policy and management for access control in OSNs, we experiment with a complex situation where a user’s content is reposted by another user in someone else’s space. Consequently, in this shared item scenario, we have multiple controllers as follows: owner, originator, contributor and multiple stakeholders, as shows in Figure 37.



Create The Scenario

First , we generate the scenario for the shared item

○ Create New Item
● Share Exist Item

Uploader User: Ahmed Saeed
Source User Ramy Mohamed
Destination User : Shady Mohamed
Item : 20
Item Content :

Tags: Kareem Mohamed | Khaled Hossam | Ayman Gamal | --NULL--

Generated Scenario : User Shared Item from another user's space to another user's space with tagged user(s)

View Policy
Controller Type : Contributor
Controller Name : Ahmed Saeed

View Policy
Controller Type : Owner
Controller Name : Shady Mohamed

View Policy
Controller Type : Originator
Controller Name : Ramy Mohamed

View Policy
Controller Type : Stakeholder
Controller Name : Kareem Mohamed
Sensetive Level :

Figure 37: The execution flow of sharing a shared item.

Similar to the previous experiment, in order to make the differences between our approach and Facebook more obvious, we introduce our results in comparison charts, in Figure 38. These charts illustrate the amount of required access control rules that have been achieved by our collaborative model and by Facebook. Our approach in all authorization types was tested against Facebook's privacy settings. The percentages of collaborative decisions in the graphs reflect the achievement of our model in term of satisfying individual co-controller policy requirements; in contrast, Facebook's percentages display the performance of its privacy settings.

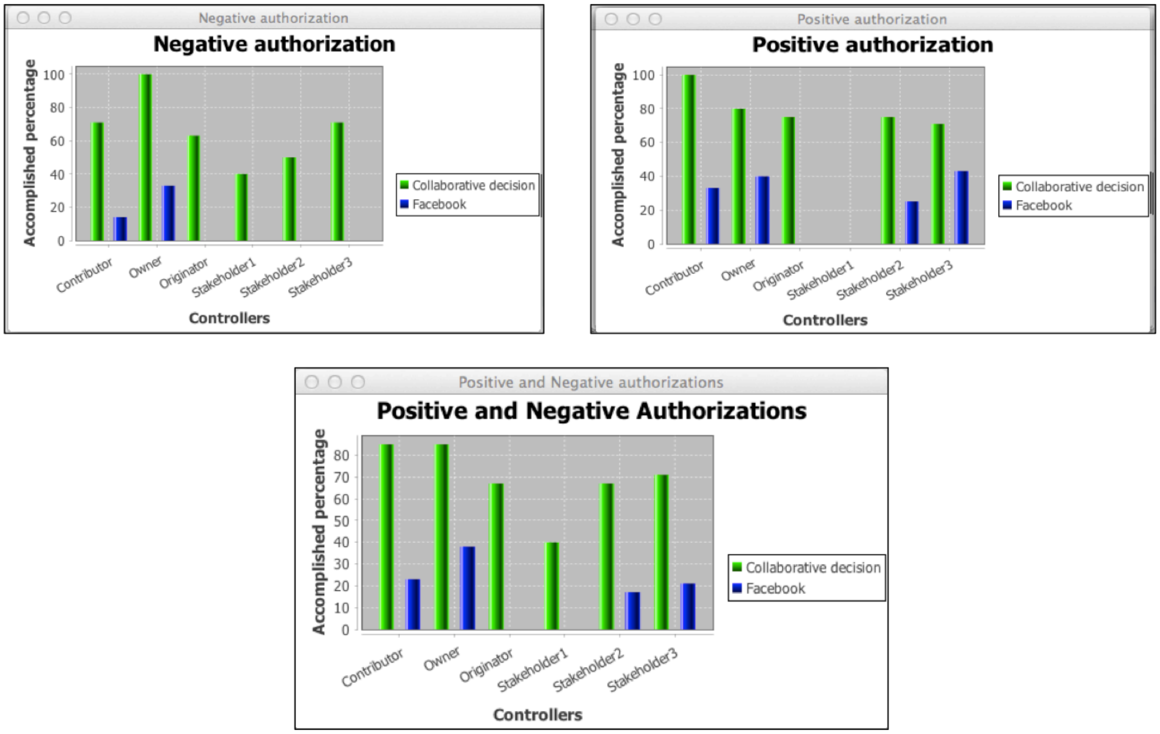


Figure 38: A comparison of the collaborative model and Facebook achievements for co-controllers privacy polices requirements in re-shared item scenarios.

Not only does the collaborative model meet high percentage of owner negative privacy desires, it promotes fairness among to the contributor, originator and stakeholders' denied privacy policies. As aforementioned, there is a high chance to have similarity between owners and other co-controllers privacy policies because our dataset is a connected graph. Subsequently, a contributor would have common privacy settings of denied accessors with the owner. Moreover, in the positive policy, there is a similarity between the owner's permitted accessors and the other associated controllers (the contributor and two of the stakeholders). As a result, Facebook in our experiment meets some co-controllers desired policies; however, the existing privacy protection mechanism in Facebook does not consider any privacy settings from other users who are respected as co-controllers in our mode. Compared to Facebook, the collaborative decision model provides more control for multiple controllers on the sharing of their content, as described in the positive and negative authorizations graph.

Overall, the results are formatted in order to measure achievement of our collaborative access control model for accessing the shared item versus Facebook. In conclusion, the prototype has shown that our proposed approach meets most of the associated controllers' privacy policies requirements, despite the challenge of maintaining a balance between the utility of social sharing and the need for privacy protection.

Chapter 6

6 Conclusion and Future Work

OSN applications are the most visited sites on the Web, where users can publish and share their personal and social information (e.g., personal data, photos, videos, opinions, contacts, etc.), as well as meet other users and electronically gather for a variety of purposes (e.g., business, entertainment, religion, education, etc.). With unexpected rapid expansion of OSN applications and the radical shift in the number of involved users around the world, OSN sites have become a rich and large repository of information about us as individuals. Additionally, OSNs are typically open systems and a valuable source of data. Due to the open nature of this huge amount of information within OSNs, which are especially important because they have been gaining popularity among Internet users, serious privacy concerns are obviously raised. Although the popularity of OSN applications has been increasing day after day, current OSNs offer primitive security mechanisms that have only limited and simple tools for controlling social network data. Several studies from different computer science disciplines and current news reports have highlighted the increase in the privacy and security issues that arise in OSNs (e.g. [Pesce, et al. 2012, Joshi and Kuo. 2011, Hongyu Gao, et al. 2011, Raad and Chbeir. 2013]).

To cope with security and privacy problems and concerns related to OSNs, we believe improving access control models is the first step. There is a general consensus that a new model of access control needs to be developed for OSNs (e.g. [Lewis, et al. 2008, Boyd and Hargittai. 2010, Cheng, et al. 2012b, Johnson, et al. 2012, Gates. 2007]). Consequently, several studies have attempted to improve and propose access control mechanisms for OSNs (e.g. [Hart, et al. 2007, Carminati, et al. 2006, 2009b, Cheng, et al. 2012a]). However, these schemes support privacy decisions as individual processes, where it is the owner of the item who is solely allowed to specify access control policies. Indeed, collaboration and sharing represent the main building blocks of OSN applications that not only are characterized by their user-driven contents, but also offer to users ingenious tools to share their personal and social information across social networks with others and take advantage of others' shared data. Typically, as a shared platform, contents in OSNs are in some sense co-owned by a number of users. OSNs have certain unique properties; in fact, sharing is one of the prominent features of existing OSN sites. Consequently, access control in OSNs presents several unique characteristics and requirements different from traditional access control models. In spite of the fact that OSNs are built on interaction, it is important to have mechanism for collaborative management of privacy settings for shared data. However, OSN applications yet do not support any mechanism for collaborative privacy management.

Although the area of OSNs has become a development part of web, access control research in this area is still in its early stages, especially the research of collaborative privacy management that is involved with multiple controllers. We believe a collaborative access control, where particular users can be considered associated controllers that all have a right to participate in the privacy management of a given shared item, plays an essential role in protecting privacy in OSNs. Accordingly, in this research, by focusing on providing methods to empower users' collaborative control over their shared items, we proposed an access control model for collaborative privacy decision making. In this Chapter, we will briefly review some details of the contributions in Section 6.1, and discuss the future work in Section 6.2.

6.1 Contributions

To designing a suitable approach to address this problem, we first need to understand scenarios of shared contents in the context of OSNs where multiple users are explicitly identified through posts, shares, tags or other metadata. Our shared items' scenarios analysis determine the set of users who have a relationship with an item by applying the social actions such as upload, share, tag or repost. Therefore, according to these relationships between users and a shared item, we distinguished the types of ownerships as follows: owner, stakeholders, contributor and originator. Moreover, based on this analysis we regulated which controllers have the right to participate in the process of making collaborative privacy decision over who can access or disseminate the shared items.

We proposed a policy specification scheme, in order to truly capture the fine-grained and collaborative authorization specifications requirements for OSN access control model. The first part of our privacy policy specification scheme is accessor specification. In reality, OSN applications target to represent the community in the real world. To keep consistency between online and offline social networks, we allow the controllers to authorize who can access their data according to accessors types. There is agreement that being relationship-based is one of the necessary requirements to establish access control systems for OSNs (e.g., [Cheng, et al. 2012a, 2012b, Carminati, et al. 2009b, 2011, Fong and Siahaan. 2011, Gates. 2007]). Also as in the real world, a user can distinguish the desirable accessors depending largely on their relationships with them; hence, we use relationship types to define authorized users in our scheme. Moreover, our relationships in our real life social network can be classified into different groups, like fun, master project, high school classmates, etc. Consequently, we also enable controllers to specify the accessors by group name where desired accessros are members in this group. Last but not least, it is very intuitive for controllers to define authorized users for their contents by user name. By three parameters in our accessor specification scheme that are user names, group names and relationship types, we support a fine-grained authorization specification requirement for OSNs. Users typically do not want to share their data with everyone; hence, we believe there is a need for specifying policies that deny access through negative authorization. By our accessor specification scheme, controllers can regulate unwelcome relationship types, groups and particular users to access their data. Indeed, negative authorization encourages users to extend their social network and to share their information with others because users can determine who cannot view their contents, instead of refusing to share their data completely, which may lead to the negative impacts on the concept of sharing that is the main purpose of OSNs. Though the combined use of positive and negative authorizations offers a precise and easy method for controllers to specify the target audience for accessing their contents, this combined with the diversified ways to enable controllers to regulate their authorized and unauthorized users do not come for free; there are conflicts. However, we applied different conflict resolution policies and some rules to solve such conflicts.

The relationships between users in the social graph are the basis that is used to specify authorized users in existing OSNs. While in the accessor specification scheme we adopted user-to-user (U2U) relationships, we believe this is not sufficient to capture our goal to propose fine-grained policy specification scheme for data and accessors. Thus, we adopt the relationship between users and resources (U2R) for the second part of our privacy policy specification scheme, that is data specification. By data specification controllers can determine the sensitivity levels of certain content that is to be shared with other users. Finally, the result of our privacy policy specification scheme is an access control policy, *p*, from each associated controller.

After each linked controller specifies individual policies, policy conflicts become inevitable. According to the understanding of collaborative privacy management requirements in OSNs and the unique characteristics that OSN access control should have, we included principles, which is the third contribution of the thesis, toward a solution for collective policy management in OSNs. The first principle is the controllers' weight scheme is driven from the reality that ownership between several controllers and particular content is not equal. Consequently, based on our controller types, we assign weights to each type, as a method to determine a priority and impact level of a controller's policy. Similarly, accessors have a different priority (weight) inferred from how they are authorized (accessors types). The accessors' weight scheme is the second principle we proposed to resolve the conflicts that occur when associated controllers have contradictory privacy preferences. In order to maintain consistency between the ways of managing online and offline social networks, we adopt trust in OSNs as a principle to resolve conflicts. Many works adopt trust between users to play a specific role in their approaches[Ali, et al. 2007, Villegas, et al. 2008, Carminati, et al. 2006, Golbeck and Hendler. 2006, Sun, et al. 2012]. We implement trust values to address the collaborative privacy issue. As in the real world, trust values between people are often associated with different levels of information disclosure; subsequently, taking a trust value into account is useful in regulating who is can access OSN resources.

As a result, these principles have been adopted to enable the collective enforcement of shared data through our algorithms where we achieve a collaborative privacy decision over who can access and share the shared contents in OSNs. These algorithms, which are the significant contribution of the thesis, are proposed for we collaborative access control model; we arrived at the three algorithms based on our multiple-user scenarios analysis which includes interactions between controllers and shared contents and the type of activities applied to these contents.

PermittedandDeniedAccessors is the essential algorithm where the final decision concerning permitted and denied sets of accessors is made. Because viewing (accessing) is different from disseminating in the context of OSNs, the AccessorSharing and ControllerSharing algorithms are formulated to reach collaborative decisions of who are allowed to disseminate the shared items. In the sharing scenarios' investigation, we observe that a dissemination action appearing from a viewers' side has different circumstances than one that comes from a controllers' side. Consequently, the AccessorSharing algorithm is articulated to produce a list of accessors, who are collaboratively authorized to disseminate the shared item with their social networks. On the other hand, the ControllerSharing algorithm is designed differently to address the policy conflicts in disseminating shared items by one of the associated controllers.

By analyzing multiple controllers' scenarios in online social networking environments, proposing a fine-grained and collaborative privacy policy specification scheme, suggesting a conflict resolution approach and designing algorithms to accomplish collaborative decisions through dissimilar privacy policies from multiple controllers who co-manage shared data, we have presented our collaborative access control model for OSNs. Achieving the desired equilibrium between providing privacy protection and the utility of sharing data in OSNs is a crucial focus of our model. Numerous recent works [Squicciarini, et al. 2009, 2010, Carminati and Ferrari. 2011, Xiao and Tan. 2012, Sun, et al. 2012, Hu, et al. 2013] have recognized the demand for multiparty management for data sharing in OSNs but, to the best of our knowledge, none of the existing approaches offer a process for making a collective decision that allows all associated users, who may be affected by the disclosure of the shared data, to setup their privacy requirements as we have done. The final contribution of the thesis is implementing our approach in a prototype. We have described a proof-of-concept implementation, and carried out the evaluation of our approach to show its feasibility and usability.

6.2 Future work

OSNs users are becoming more aware of shared data privacy because, as in the real world, users do not want to share their data with everyone all the time. Consequently, one of the fundamental requirements for OSNs users is to enable them to specify the audience and enforce their access control policy. When this requirement comes from different parties, who co-own the shared item and may be affected by disclosure of it, providing an efficient access control model over the information shared is a complicated task, especially, with the vast number of users in OSNs and the tremendous amount of shared data. Thus, there are a number of directions that the work presented here can be extended and improved.

While this work offers a model to capture the essence of collaborative authorization requirements for data sharing in OSNs, we need to enhance our model to work appropriately in an online platform where algorithms have to run online for such online problem. Relationships play a significant part when improving the paradigm of access control for OSNs, which are based on relationships. For this reason, we will support our privacy policy specification scheme by adding transitive relationships with length 2 (*R.of.R*). Moreover, to handle several characteristics that relationships in OSNs should have, as part of future work, we will develop our scheme to offer additional characteristics of relationships to improve our collaborative access control model; for example, supporting multiple relationships that exist between two users (e.g., both co-worker and friend). Also, adding one-direction relationships (e.g., a parent, daughter, son, manger, etc.) might be considered to be one direction relationships and other various relationship features such as its content, and strength is an interesting future direction for an OSN collaborative access control model.

In our model, we assume that there is not an identical trust level between each two users who are in any type of relationships. We would extend our trust network to cover distrust property because propagation of trust and distrust values along a social network graph allows a user to form more validated trust value on a user who is not directly connected to her/him.

Although OSNs are evolving, the topic of collaborative access control is not fully explored in the research literature. Therefore, we are planning to investigate and analyze more about multiple controllers' scenarios in OSNs. For example, some of our sharing scenarios such as Figures 13 and 14 where an intermediate user shares other users' content and posts it in someone else's space, we need to verify who is the new profile's owner and if disclosing content to this new owner and her/his social network may affect the privacy of the previous owner and stakeholders. In our approach for sharing cases, the privacy policies specified by co-controllers are collaboratively enforced to regulate the final decision of accessing the item in a new owner's space; however, the shared item may be further re-disseminated by users who belong to the new owner's social network. Hence, generalizing and enhancing the access control model to cope with the shared item regardless of how many times it has been re-disseminated (reposted) is an interesting future direction.

According to statistics from most popular OSN's sites, photo sharing and tagging are an extremely popular activity and tagged users are explicitly recognized through tags; however, the permissions from other users appearing in the photo is not required. An effective idea to overcome this limitation is the face detection and face recognition techniques. Consequently, we would extend our work to have them automatically tagged by using these techniques [Shih-Chia Huang, et al. 2014, Hsu, et al. 2013, Choi, et al. 2011]. Another automatic idea is to incorporate inference-based techniques for semiautomated generation of access control polices [Squicciarini, et al. 2014, 2011b]. The idea behind integrating this technique into an OSN's access control model is that generally specifying the privacy policies may be tedious and perplexing for the ordinary user with large amounts of personal data and hundreds of connections on OSN platforms. Additionally, in collaborative privacy management, more involvements from other associated controllers is required. Consequently, privacy settings become a more difficult and complicated task. Therefore, supporting semiautomated generation of access control rules seems to be the most critical aspect of developing collaborative access control mechanisms in the future. Finally, we would to build an application as a proof-of-concept for our approach and show and analyze the effectiveness of our proposal by doing experiments on diverse datasets.

References

ACQUISTI, A. AND GROSS, R. 2006. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In Springer Berlin Heidelberg, Berlin, Heidelberg, 36-58.

AIMEUR, E., GAMBS, S. AND AI HO. 2009. UPP: User Privacy Policy for Social Networking Sites. In *Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on,* 267-272.

AKCORA, C.G., CARMINATI, B. AND FERRARI, E. 2012. Risks of friendships on social networks. *arXiv preprint arXiv:1210.3234* .

ALI, B., VILLEGAS, W. AND MAHESWARAN, M. 2007. A trust based approach for protecting user data in social networks. In *Proceedings of the 2007 conference of the center for advanced studies on Collaborative research,* IBM Corp., 288-293.

ANDERSON, J. AND STAJANO, F. 2013. Must Social Networking Conflict with Privacy? *IEEE Security & Privacy* 11, 51-60.

ANWAR, M. AND FONG, P.W. 2012. A visualization tool for evaluating access control policies in facebook-style social network systems. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing,* ACM, 1443-1450.

ANWAR, M., ZHAO, Z. AND FONG, P.W. 2010. An access control model for Facebook-style social network systems.*University of Calgary*.

BALDWIN, R.W. 1990. Naming and grouping privileges to simplify security management in large databases. In *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on,* IEEE, 116-132.

BARRET, D. AND SAUL, M.H. 2011. Weiner now says he sent photos. *The Wall Street Journal*.

BECKER, J.L. 2009. Measuring privacy risk in online social networks. ProQuest, UMI Dissertations Publishing.

BELL, D.E. AND LA PADULA, L.J. 1976. Secure computer system: Unified exposition and multics interpretation. DTIC Document.

BENANTAR, M. 2006. Access control systems: security, identity management and trust models. Springer Verlag, DE.

BERTINO, E., BONATTI, P.A. AND FERRARI, E. 2001. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC)* 4, 191-233.

BERTINO, E., BUCCAFURRI, F., FERRARI, E. AND RULLO, P. 2000. A Logic-based Approach for Enforcing Access Control[1]A Preliminary Version of This Paper Appears in the Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS\&Rsquo;98), Louvain-La-Neuve, Belgium, September 1998 Under the Title \&Ldquo;An Authorization Model and Its Formal Semantics\&Rdquo;. *J.Comput.Secur.* 8, 109-139. http://dl.acm.org/citation.cfm?id=1297828.1297831.

BERTINO, E. AND SANDHU, R. 2005. Database security-concepts, approaches, and challenges. *Dependable and Secure Computing, IEEE Transactions on* 2, 2-19.

BESMER, A. AND RICHTER LIPFORD, H. 2010. Moving Beyond Untagging: Photo Privacy in a Tagged World. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* Atlanta, Georgia, USA, ACM, New York, NY, USA, 1563-1572.

BIBA, K.J. 1977. Integrity considerations for secure computer systems. ESD-TR-76-372, USAF Electronic Systems Division, Bedford, Mass.

BLAZE, M., FEIGENBAUM, J. AND LACY, J. 1996. Decentralized trust management. In *Security and Privacy, 1996. Proceedings.,1996 IEEE Symposium on,* IEEE, 164-173.

BONATTI, P.A. AND OLMEDILLA, D. 2007. Rule-based policy representation and reasoning for the semantic web. In *Reasoning Web,* Springer, 240-268.

BONATTI, P.A., DE COI, J.L., OLMEDILLA, D. AND SAURO, L. 2009. Rule-based policy representations and reasoning. In *Semantic techniques for the web,* Springer, 201-232.

BOYD, D.M. AND ELLISON, N.B. 2007. Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication* 13, 210-230.

BOYD, D. AND HARGITTAI, E. 2010. Facebook privacy settings: Who cares? *First Monday* 15.

BRAND, S.L. 1985. DoD 5200.28-STD Department of Defense Trusted Computer System Evaluation Criteria (Orange Book). *National Computer Security Center* 1-94. .

BRUNS, G., FONG, P.W., SIAHAAN, I. AND HUTH, M. 2012. Relationship-based access control: its expression and enforcement through hybrid logic. In *Proceedings of the second ACM conference on Data and Application Security and Privacy,* ACM, 117-124.

CAMBRIDGE DICTIONARIES ONLINE. 2014. English definition of "trust". 2014, http://dictionary.cambridge.org/us/dictionary/british/trust.

CARLEY, K. 1991. A Theory of Group Stability. *American Sociological Review* 56, 331-354.

CARMINATI, B., FERRARI, E. AND VIVIANI, M. 2014. Security and trust in online social networks. San Rafael, California (1537 Fourth Street, San Rafael, CA 94901 USA) : Morgan & Claypool, 2014 .

CARMINATI, B. AND FERRARI, E. 2011. Collaborative access control in on-line social networks. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on,* IEEE, 231-240.

CARMINATI, B. AND FERRARI, E. 2008. Privacy-Aware Collaborative Access Control in Web-Based Social Networks. In Springer Berlin Heidelberg, Berlin, Heidelberg, 81-96.

CARMINATI, B., FERRARI, E., HEATHERLY, R., KANTARCIOGLU, M. AND THURAISINGHAM, B. 2011. Semantic web-based social network access control. *Computers & security* 30, 108-115.

CARMINATI, B., FERRARI, E., HEATHERLY, R., KANTARCIOGLU, M. AND THURAISINGHAM, B. 2009a. A semantic web based framework for social network access control. In *Proceedings of the 14th ACM symposium on Access control models and technologies,* ACM, 177-186.

CARMINATI, B., FERRARI, E. AND PEREGO, A. 2009b. Enforcing access control in web-based social networks. *ACM Transactions on Information and System Security (TISSEC)* 13, 1, 1-38

CARMINATI, B., FERRARI, E. AND PEREGO, A. 2008. A decentralized security framework for web-based social networks. *International Journal of Information Security and Privacy (IJISP)* 2, 22-53.

CARMINATI, B., FERRARI, E. AND PEREGO, A. 2007. Private relationships in social networks. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on,* IEEE, 163-171.

CARMINATI, B., FERRARI, E. AND PEREGO, A. 2006. Rule-Based Access Control for Social Networks. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops,* R. MEERSMAN, Z. TARI AND P. HERRERO, Eds. Springer Berlin Heidelberg, 1734-1744.

CAVERLEE, J., LIU, L. AND WEBB, S. 2008. Socialtrust: tamper-resilient trust establishment in online communities. In ACM, 104-114.

CHENG, Y., PARK, J. AND SANDHU, R. 2012a. Relationship-based access control for online social networks: Beyond user-to-user relationships. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom),* IEEE, 646-655.

CHENG, Y., PARK, J. AND SANDHU, R. 2012b. A user-to-user relationship-based access control model for online social networks. In *Data and applications security and privacy XXVI,* Springer, 8-24.

CHI ZHANG, JINYUAN SUN, XIAOYAN ZHU AND YUGUANG FANG. 2010. Privacy and security for online social networks: challenges and opportunities. *Network, IEEE* 24, 13-18.

CHIU, P., CHEUNG, C.M.K. AND LEE, M.K.O. 2008. Online Social Networks: Why Do "We" Use Facebook? In Springer Berlin Heidelberg, Berlin, Heidelberg, 67-74.

CHOI, J.Y., NEVE, W.D., PLATANIOTIS, K.N. AND RO, Y.M. 2011. Collaborative face recognition for improved face annotation in personal photo collections shared on online social networks. *Multimedia, IEEE Transactions on* 13, 14-28.

DEEP NISHAR. April 18, 2014. The Next Three Billion [INFOGRAPHIC]. *Linkedin Official Blog* April 2014. http://blog.linkedin.com/2014/04/18/the-next-three-billion/.

DEUTSCH, M. 1962. Cooperation and trust: Some theoretical notes. In *M. R. Jones (Ed.), Nebraska symposium on motivation (Vol. X)*. Lincoln: University of Nebraska Press.

DI VIMERCATI, S., FORESTI, S., SAMARATI, P. AND JAJODIA, S. 2007. Access control policies and languages. *International Journal of Computational Science and Engineering* 3, 94-102.

DI VIMERCATI, S., SAMARATI, P. AND JAJODIA, S. 2005. Policies, models, and languages for access control. SPRINGER-VERLAG BERLIN, BERLIN, 225-237.

EXPERIAN MARKETING SERVICES. 2012. The 2012 Digital Marketer: Benchmark and Trend Report . December 2013. http://go.experian.com/forms/experian-digital-marketer-2012**.**

FACEBOOK. September 2014. Company Info. September 2014. http://newsroom.fb.com/company-info/.

FACEBOOK, ERICSSON AND QUALCOMM. 2013. A Focus on Efficiency. 15.JUN 2014. https://fbcdn-dragon-a.akamaihd.net/hphotos-ak ash3/851590_229753833859617_1129962605_n.pdf.

FELD, S.L. 1981. The focused organization of social ties. *American journal of sociology* 1015-1035.

FERRAIOLO, D.F. AND KUHN, D.R. 2009. Role-based access controls. *arXiv preprint arXiv:0903.2171* .

FERRAIOLO, D.F., SANDHU, R., GAVRILA, S., KUHN, D.R. AND CHANDRAMOULI, R. 2001. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)* 4, 224-274.

FERRARI, E. 2010. Access Control in Data Management Systems. Morgan & Claypool Publishers.

FERRARI, E. 2009. Access Control. In *Encyclopedia of Database Systems,* Springer US, 7-11.

FERRARI, E. AND THURAISINGHAM, B. 2000. Secure database systems. *Advanced databases: technology and design.Artech House* .

FONG, P.W.L. 2011a. Preventing Sybil Attacks by Privilege Attenuation: A Design Principle for Social Network Systems. In IEEE, 263-278.

FONG, P.W. 2011b. Relationship-based access control: protection model and policy language. In *Proceedings of the first ACM conference on Data and application security and privacy,* ACM, 191-202.

FONG, P.W. AND SIAHAAN, I. 2011. Relationship-based access control policies and their policy languages. In *Proceedings of the 16th ACM symposium on Access control models and technologies,* ACM, 51-60.

FONG, P.L., ANWAR, M. AND ZHAO, Z. 2009. A Privacy Preservation Model for Facebook-Style Social Network Systems. In *Computer Security â€“ ESORICS 2009,* M. BACKES AND P. NING, Eds. Springer Berlin Heidelberg, 303-320.

GAIL-JOON, A., SHEHAB, M. AND SQUICCIARINI, A. 2011. Security and Privacy in Social Networks. *Internet Computing, IEEE* 15, 10-12.

GATES, C.E. 2007. Access control requirements for web 2.0 security and privacy. In *Proc. of Workshop on Web 2.0 Security and Privacy Workshop (W2SP 2007),* Citeseer.

GOEL, S., HOFMAN, J.M. AND SIRER, M.I. 2012. Who Does What on the Web: A Large-Scale Study of Browsing Behavior. In *ICWSM.*

GOLBECK, J. 2006. Trust on the world wide web: a survey. *Foundations and Trends in Web Science* 1, 131-197.

GOLBECK, J.A. 2005. Computing and applying trust in web-based social networks. ProQuest, UMI Dissertations Publishing.

GOLBECK, J. AND HENDLER, J. 2006. Inferring binary trust relationships in Web-based social networks. *ACM Transactions on Internet Technology (TOIT)* 6, 497-529.

GOLLU, K.K., SAROIU, S. AND WOLMAN, A. 2007. A social networking-based access control scheme for personal content. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP'07)-Work-in-Progress Session*.

GOOGLE OFFICIAL BLOG. April 11, 2012. February 2014.
http://googleblog.blogspot.fr/2012/04/toward-simpler-more-beautiful-google.html.

GROSS, R. AND ACQUISTI, A. 2005. Information Revelation and Privacy in Online Social Networks. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society,* Alexandria, VA, USA, ACM, New York, NY, USA, 71-80.

GROSSKLAGS, J., CHRISTIN, N. AND CHUANG, J. 2008. Secure or insure?: a game-theoretic analysis of information security games. In *Proceedings of the 17th international conference on World Wide Web,* ACM, 209-218.

GURSES, S. AND DIAZ, C. 2013. Two tales of privacy in online social networks. *Security & Privacy, IEEE* 11, 29-37.

GUTTMAN, B. AND ROBACK, E.A. 1995. An introduction to computer security: the NIST handbook. DIANE Publishing.

HAMPTON, K., GOULET, L.S., RAINIE, L. AND PURCELL, K. 2011. Social networking sites and our lives. http://pewinternet.org/Reports/2011/Technology-and-social-networks.aspx.

HART, M., JOHNSON, R. AND STENT, A. 2007. More content-less control: Access control in the web 2.0. *IEEE Web* 2.

HENNE, B., LINKE, M. AND SMITH, M. 2014. A Study on the Unawareness of Shared Photos in Social Network Services. In *IEEE Security and Privacy Workshop on Web 2.0 Security & Privacy 2014 (W2SP),* San Jose, California, May 18 2014, IEEE.

HONGYU GAO, JUN HU, TUO HUANG, JINGNAN WANG AND YAN CHEN. 2011. Security Issues in Online Social Networks. *Internet Computing, IEEE* 15, 56-63.

HOWARD, B. 2008. Analyzing online social networks. *Communications of the ACM* 51, 14-16.

HSU, C., JIAU, M. AND HUANG, S. 2013. An Automatic Face Annotation System Featuring High Accuracy for Online Social Networks. In *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC),* IEEE, 163-169.

HU, H. AND AHN, G. 2011. Multiparty authorization framework for data sharing in online social networks. In *Data and Applications Security and Privacy XXV,* Springer, 29-43.

HU, H., AHN, G. AND JORGENSEN, J. 2013. Multiparty Access Control for Online Social Networks: Model and Mechanisms. *Knowledge and Data Engineering, IEEE Transactions on* 25, 1614-1627.

HU, H., AHN, G. AND JORGENSEN, J. 2012. Enabling collaborative data sharing in google+. In *Global Communications Conference (GLOBECOM), 2012 IEEE,* 720-725.

HU, H., AHN, G. AND JORGENSEN, J. 2011. Detecting and resolving privacy conflicts for collaborative data sharing in online social networks. In *Proceedings of the 27th Annual Computer Security Applications Conference,* ACM, 103-112.

INCITS, A. 2004. INCITS 359-2004. *Role based access control* .

ISO, I. 1989. 7498-2. Information Processing Systems Open Systems Interconnection Basic Reference Model-Part 2: Security Architecture. *ISO Geneva, Switzerland* .

JAJODIA, S., SAMARATI, P., SAPINO, M.L. AND SUBRAHMANIAN, V. 2001. Flexible support for multiple access control policies. *ACM Transactions on Database Systems (TODS)* 26, 214-260.

JIN, L., JOSHI, J. AND ANWAR, M. 2013. Mutual-friend based attacks in social network systems. *Computers & Security* 37, 15-30.

JOHNSON, M., EGELMAN, S. AND BELLOVIN, S. 2012. Facebook and privacy: it's complicated. In ACM, 1-15.

JØSANG, A., ISMAIL, R. AND BOYD, C. 2007. A survey of trust and reputation systems for online service provision. *Decision Support Systems* 43, 618-644.

JOSHI, J.B., BERTINO, E., LATIF, U. AND GHAFOOR, A. 2005. A generalized temporal role-based access control model. *Knowledge and Data Engineering, IEEE Transactions on* 17, 4-23 .

JOSHI, P. AND KUO, C.-.J. 2011. Security and privacy in online social networks: A survey. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on,* 1-6.

KAIRAM, S., BRZOZOWSKI, M., HUFFAKER, D. AND CHI, E. 2012. Talking in circles: selective sharing in google+. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* ACM, 1065-1074.

KIM, S. AND HAN, S. 2009. The method of inferring trust in web-based social network using fuzzy logic. In *international workshop on machine intelligence research,* 140-144.

KIM, W., JEONG, O. AND LEE, S. 2010. On social Web sites. *Information Systems* 35, 215-236. http://resolver.scholarsportal.info/resolve/03064379/v35i0002/215_osws.

KRUK, S.R. 2004. FOAF-Realm-control your friends' access to the resource. In *FOAF Workshop proceedings*.

KRUK, S., GRZONKOWSKI, S., GZELLA, A., WORONIECKI, T. AND CHOI, H. 2006. D-FOAF: Distributed identity management with access rights delegation. In SPRINGER-VERLAG BERLIN, BERLIN, 140-154.

KUTER, U. AND GOLBECK, J. 2007. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In *AAAI*, 1377-1382.

LAMPE, C., ELLISON, N. AND STEINFIELD, C. 2006. A face(book) in the crowd: social Searching vs. social browsing. In ACM, 167-170.

LAMPSON, B.W. 1974. Protection. *SIGOPS Oper.Syst.Rev.* 8, 18-24. http://doi.acm.org/10.1145/775265.775268.

LESANI, M. AND MONTAZERI, N. 2009. FUZZY TRUST AGGREGATION AND PERSONALIZED TRUST INFERENCE IN VIRTUAL SOCIAL NETWORKS. *Computational Intelligence* 25, 51-83. .

LESANI, M. AND BAGHERI, S. 2006. Applying and Inferring Fuzzy Trust in Semantic Web Social Networks. In Springer US, Boston, MA, 23-43.

LEVY, H.M. 1984. Capability-based computer systems. Digital Press Bedford.

LEWIS, K., KAUFMAN, J. AND CHRISTAKIS, N. 2008. The taste for privacy: An analysis of college student privacy settings in an online social network. *Journal of Computer-Mediated Communication* 14, 79-100.

LI, N., GROSOF, B.N. AND FEIGENBAUM, J. 2003. Delegation logic: A logic-based approach to distributed authorization. *ACM Transactions on Information and System Security (TISSEC)* 6, 128-171.

LI, Y., LI, Y., YAN, Q. AND DENG, R.H. 2013. Think Twice before You Share: Analyzing Privacy Leakage under Privacy Control in Online Social Networks. In *Network and System Security,* Springer, 671-677.

LIU, Y., GUMMADI, K.P., KRISHNAMURTHY, B. AND MISLOVE, A. 2011. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference,* ACM, 61-70.

LOUKIDES, G. AND GKOULALAS-DIVANIS, A. 2009. Privacy challenges and solutions in the social web. *Crossroads* 16, 14-18.

LUHMANN, N. 1979. Trust and power Chichester. *United Kingdom: John Wiley and Sons, Inc*.

LUNT, T.F. 1989. Access control policies: Some unanswered questions. *Computers & Security* 8, 43-54.

MADDEN, M. 2012. Privacy management on social media sites. *Pew Internet Report* 1-20.

MADEJSKI, M., JOHNSON, M. AND BELLOVIN, S.M. 2012. A study of privacy settings errors in an online social network. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on,* 340-345.

MAHMOOD, S. AND DESMEDT, Y. 2012. Your Facebook deactivated friend or a cloaked spy. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on,* 367-373.

MAHMOOD, S. 2013. Online Social Networks: Privacy Threats and Defenses. In *Security and Privacy Preserving in Social Networks,* Springer, 47-71.

MASOUMZADEH, A. AND JOSHI, J. 2013. Privacy settings in social networking systems: what you cannot control. In ACM, 149-154.

MASOUMZADEH, A. AND JOSHI, J. 2011. Ontology-based access control for social network systems. *International Journal of Information Privacy, Security and Integrity* 1, 59-78.

MISLOVE, A., MARCON, M., GUMMADI, K.P., DRUSCHEL, P. AND BHATTACHARJEE, B. 2007. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement,* ACM, 29-42.

MISLOVE, A., VISWANATH, B., GUMMADI, K. AND DRUSCHEL, P. 2010. You are who you know: inferring user profiles in online social networks. In ACM, 251-260.

MOYER, M.J. AND ABAMAD, M. 2001. Generalized role-based access control. In *Distributed Computing Systems, 2001. 21st International Conference on.,* 391-398.

MUSIAÅ, K. AND KAZIENKO, P. 2013. Social networks on the Internet. *World Wide Web* 16, 31-72. http://dx.doi.org/10.1007/s11280-011-0155-z.

NGENO, C., ZAVARSKY, P., LINDSKOG, D. AND RUHL, R. 2010. User's Perspective: Privacy and Security of Information on Social Networks. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on,* 1038-1043.

NIELSEN. 2012. State of the Media: The Social Media Report. *The Nielsen Company* http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2012-Reports/The-Social-Media-Report-2012.pdf.

NYANCHAMA, M. AND OSBORN, S. 1999. The role graph model and conflict of interest. *ACM Transactions on Information and System Security (TISSEC)* 2, 3-33.

NYANCHAMA, M. AND OSBORN, S.L. 1994. Access Rights Administration in Role-Based Security Systems. In *DBSec,* Citeseer, 37-56.

PALLIS, G., ZEINALIPOUR-YAZTI, D. AND DIKAIAKOS, M.D. 2011. Online Social Networks: Status and Trends. In Springer Berlin Heidelberg, Berlin, Heidelberg, 213-234.

PANG, J. AND ZHANG, Y. 2013. A New Access Control Scheme for Facebook-style Social Networks. *arXiv preprint arXiv:1304.2504* .

PESCE, J., CASAS, D., RAUBER, G. AND ALMEIDA, V. 2012. Privacy attacks in social media using photo tagging networks: a case study with Facebook. In ACM, 1-8.

RAAD, E. AND CHBEIR, R. 2013. Privacy in Online Social Networks. In *Security and Privacy Preserving in Social Networks,* Springer, 3-45.

RITCHIE, D.M. AND THOMPSON, K. 1983. The UNIX Time-sharing System. *Commun.ACM* 26, 84-89. http://doi.acm.org/10.1145/357980.358014.

RYUTOV, T., KICHKAYLO, T. AND NECHES, R. 2009. Access Control Policies for Semantic Networks. In *Policies for Distributed Systems and Networks, 2009. POLICY 2009. IEEE International Symposium on,* July, 150-157.

SAMARATI, P. AND DE VIMERCATI, S.C. 2001. Access control: Policies, models, and mechanisms. In *Foundations of Security Analysis and Design,* Springer, 137-196.

SANDHU, R.S., COYNE, E.J., FEINSTEIN, H.L. AND YOUMAN, C.E. 1996. Role-based access control models. *Computer* 29, 38-47.

SANDHU, R.S. AND SAMARATI, P. 1994. Access control: principle and practice. *Communications Magazine, IEEE* 32, 40-48.

SCHLENKER, B.R., HELM, B. AND TEDESCHI, J.T. 1973. The effects of personality and situational variables on behavioral trust. *Journal of personality and social psychology* 25, 419-427.

SCHNEIDER, F., FELDMANN, A., KRISHNAMURTHY, B. AND WILLINGER, W. 2009. Understanding online social network usage from a network perspective. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference,* ACM, 35-48.

SHERCHAN, W., NEPAL, S. AND PARIS, C. 2013. A survey of trust in social networks. *ACM Computing Surveys (CSUR)* 45, 1-33.

SHIH-CHIA HUANG, MING-KAI JIAU AND CHIH-AN HSU. 2014. A High-Efficiency and High-Accuracy Fully Automatic Collaborative Face Annotation System for Distributed Online Social Networks. *Circuits and Systems for Video Technology, IEEE Transactions on* 24, 1800-1813.

SOCIALBAKERS. Facebook Statistics by Country. February 2014. http://www.socialbakers.com/all-social-media-stats/facebook/.

SQUICCIARINI, A.C., PACI, F. AND SUNDARESWARAN, S. 2014. PriMa: a comprehensive approach to privacy protection in social network sites. *annals of telecommunications-annales des télécommunicationss* 69, 21-36.

SQUICCIARINI, A.C., SHEHAB, M. AND PACI, F. 2009. Collective privacy management in Social networks, 521-530.

SQUICCIARINI, A.C., SHEHAB, M. AND WEDE, J. 2010. Privacy policies for shared content in social network sites. *The VLDB Journal* 19, 777-796.

SQUICCIARINI, A.C., XU, H. AND ZHANG, X.L. 2011a. CoPE: Enabling collaborative privacy management in online social networks. *Journal of the American Society for Information Science and Technology* 62, 521-534.

SQUICCIARINI, A.C., SUNDARESWARAN, S., LIN, D. AND WEDE, J. 2011b. A3p: adaptive policy prediction for shared images over popular content sharing sites. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia,* ACM, 261-270.

STALLINGS, W. 2008. Computer security: principles and practice. Prentice Hall, Upper Saddle River, NJ.

STUTZMAN, F. 2006. An evaluation of identity-sharing behavior in social network communities. *International Digital and Media Arts Journal* 3, 10-18.

SUN, Y., ZHANG, C., PANG, J., ALCALDE, B. AND MAUW, S. 2012. A trust-augmented voting scheme for collaborative privacy management. *Journal of Computer Security* 20, 437-459.

TWITTER INC. June 2014. Twitter usage. June 2014. https://about.twitter.com/company.

VAIDYA, J., CLIFTON, C.W. AND ZHU, M. 2006. Privacy preserving data mining. Springer, New York.

VILLEGAS, W., ALI, B. AND MAHESWARAN, M. 2008. An access control scheme for protecting personal data. In *Privacy, Security and Trust, 2008. PST'08. Sixth Annual Conference on,* IEEE, 24-35.

VORAKULPIPAT, C., MARKS, A., REZGUI, Y. AND SIWAMOGSATHAM, S. 2011. Security and privacy issues in Social Networking sites from user's viewpoint. In *Technology Management in the Energy Smart World (PICMET), 2011 Proceedings of PICMET '11*, 1-4.

WASSERMAN, S. AND FAUST, K. 1997. Social network analysis: methods and applications. Cambridge University Press, New York; Cambridge [England].

WEEKS, S. 2001. Understanding trust management systems. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on,* IEEE, 94-105.

WISHART, R., CORAPI, D., MARINOVIC, S. AND SLOMAN, M. 2010. Collaborative privacy policy authoring in a social networking context. In *Policies for Distributed Systems and Networks (POLICY), 2010 IEEE International Symposium on,* IEEE, 1-8.

XIAO, Q. AND TAN, K. 2012. Peer-aware collaborative access control in social networks. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on,* 30-39.

XUE, M., CARMINATI, B. AND FERRARI, E. 2011. P3d-privacy-preserving path discovery in decentralized online social networks. In *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual,* IEEE, 48-57.

YEUNG, C.A., LICCARDI, I., LU, K., SENEVIRATNE, O. AND BERNERS-LEE, T. 2009. Decentralization: The future of online social networking. In *W3C Workshop on the Future of Social Networking Position Papers,* 2-7.

YOUNG, H.P. 1988. Condorcet's Theory of Voting. *The American Political Science Review* 82, 1231-1244.

Curriculum Vitae

Name:	Hanaa Alshareef
Post-secondary Education and Degrees:	Umm Al Qura University Makkah, Saudi Arabia 2004 -2008 B.A.
Related Work Experience	Teaching Assistant King Khalid University 2008-2010