Thesis for The Degree of Licentiate of Engineering

A Collaborative Access Control Framework for Online Social Networks

Hanaa Alshareef

CHALMERS

Division of Formal Methods
Department of Computer Science & Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2019

A Collaborative Access Control Framework for Online Social Networks

Hanaa Alshareef

Technical Report No Technical report 196L
ISSN 1652-876X
Department of Computer Science & Engineering
Division of Formal Methods
Chalmers University of Technology
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2019.

Abstract

Online social networks (OSNs) are one of the most popular web-based services for people to communicate and share information with each other. With all their benefits, OSNs might raise serious problems in what concerns users' privacy. One privacy risk is caused by accessing and sharing co-owned data items, i.e., when a user posts a data item that involves other users, some users' privacy may be disclosed, since users generally have different privacy preferences regarding who can access and share their data. Another risk is caused by the privacy settings offered by OSNs that do not, in general, allow fine-grained enforcement, especially in cases where posted data items concern other users. We discuss and give examples of these issues, in order to illustrate their impacts on current OSNs' privacy protection mechanisms. We propose a collaborative access control framework to deal with such privacy issues. Basically, in our framework, the decision whether a user can access or share a co-owned data item is based on the aggregated opinion of all users involved. Our solution is based on the sensitivity level of users with respect to the concerned data item, the trust among users, the types of *controllers* (those who are concerned in making the collaborative decision) and the types of *accessors* (those who are identified to access a given data item or not). In order to observe how varying some of the parameters mentioned above influence the outcome of the permitting/denying decision of the proposed solution, we provide an evaluation of our framework. We also present a proof-of-concept implementation of our approach in the open source OSN Diaspora.

Keywords: Collaborative Access Control, Multiparty, Privacy, Online Social Networks

Acknowledgments

It is a pleasure for me to acknowledge the support of many people throughout my journey. Firstly, I would like to express my sincere and profound gratitude to my supervisor Gerardo, for his unwavering support, encouragement, guidance, thoughtful advice and insightful comments that have been helpful to improve and advance my knowledge.

I am also grateful to Dave and Wolfgang for their help in the past two years and for the interesting discussions about science, history, or food.

I am beholden to many whom by being around, have helped me to settle into welcoming surroundings and made my time here more pleasant. Thank you all! Raúl and Pablo, for the fruitful discussions and insightful comments; Mauricio for being a good friend and officemate and making the work place a fun space; Iulia for standing as inspiring model of women in computer science; Shirin for being such a good listener and supportive. Thank you, everyone who is making Chalmers such a welcoming and friendly environment.

My deepest gratitude is directed towards my family, for your unconditional love and countless support. A big thanks goes to my friends, Wafaa ALshareef, Muwada, Tagreed, Azhar, Mohammad and Travis who were the closest to me all over the way and still there, forgave much, put up with much and loved much.

Contents

Introduction **1**
1 Privacy . . . 1
2 Online Social Networks . . . 2
2.1 Privacy Policies in Online Social Networks . . . 3
2.2 Access Control in Online Social Networks . . . 11
3 Thesis Overview . . . 14
Bibliography . . . 15

1 A Collaborative Access Control Framework for Online Social Networks **21**
1 Introduction . . . 23
2 A Collaborative Access Control Framework for OSNs . . . 25
2.1 OSN Model . . . 25
2.2 Access Control Policies . . . 29
2.3 Collaborative Access Control . . . 34
2.4 Computational Complexity . . . 39
3 Evaluation . . . 41
3.1 Proof-Of-Concept Implementation . . . 49
4 Related Work . . . 51
5 Conclusion . . . 53
Bibliography . . . 55
Appendix . . . 59

Introduction

1 Privacy

Although there is a prevailing belief that privacy is quite a new concept, developed with the capability of new web-services technologies, this view is not quite precise. During the early time of the industrial revolution, officials perceived privacy as a default regulation of human life. The right of privacy that emerged during the Gilded Age (1840-1950), was formed into a constitutional creed by 1965, which considered the oldest constitutional rights. Warren and Brandeis define privacy as the "right to be let alone" [60]. Decades later, Westin referred to it as "the clime of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others." [61]. Because of the importance of privacy for an individual's autonomy, identity, and integrity many have attempted to define privacy [3, 61]. However, the criticism on the complexity and ambiguity of such definitions is still quite prevalent [57].

Privacy is recognized as a human right by many international and regional agreements, such as the Universal Declaration of Human Rights [5, 40]. Privacy is closely related to other fundamental human rights such as dignity, personal autonomy or self-determination, freedom, individuality, respect, etc. This gives privacy paramount importance.

In the twenty-first century, technological change has apparently shaken up society and privacy along with it; thus, theories and models have been developed to achieve and meet the demands of this change. Information privacy has firstly and clearly been realized as an issue when the internet was entirely commercialized in the United States [21]. Hence, the Data Protection Directive of the European Union has defined information privacy explicitly as a basic human right [24].

Given the dramatic improvements in information technologies (e.g., Big Data, digital identity, biometrics and online social networks) in the last decade, and the increasing processing and storing capacity of computer devices, technology has become pervasive in our daily activities. A massive amount of information is thus available over those technologies, making privacy particularly important in the socio-technical landscape. The proliferation of online data collected in everyday life has a destructive effect on privacy due to the sensitivity of the data collected and shared without convenient control or monitoring.

Online services, such as online social networks, provide immense benefit for the entire society. However, they have also created lots of unanticipated privacy breaches that compromise individual privacy. In the following section, we briefly explain the structure of online social networks and its possible privacy breaches.

2 Online Social Networks

With the increasing popularity of the World Wide Web (WWW), many different web-based services become available, including *online social networks (OSNs)*. OSNs promote online social interactions between individuals, for instance making a relationship, meeting others and sharing information [36]. As reported by Boyd and Ellison [10], OSNs are distinguished from other web-based services by three characteristics: first, they have a public or semi-public *profile*, which is a web-page that describes the user by information such as age, location or interests among others; second, the profile contains a set of connections or relationships between users might be established based on suggestion from the OSN, relying on the public's personal information such as name, location, birthday, personal interests, etc. to structure them. Third, OSNs provide users with the ability to share and view certain information (e.g., photos, contacts, interests, activities, backgrounds, etc.) about others they are linked to.

Since their introduction, OSNs like Facebook, Twitter, Instagram and LinkedIn have attracted over two billion users, who are uploading and sharing hundreds of billions of data items through them. And as they are getting further integrated in the daily life of many more people, the number of users and amount of data they would be uploading and sharing are expected to continue growing in the coming years.

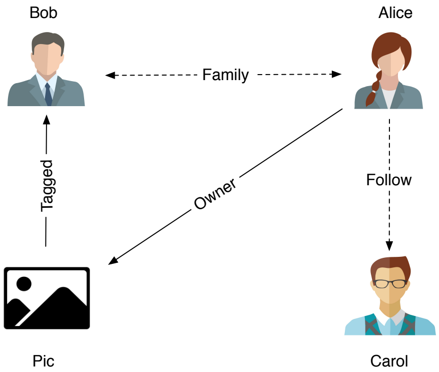


Figure 1: Social Graph Example

Given the inherent structure of OSNs, the most common way to represent OSNs are *graphs* (usually called *social graphs* in this context) [11]. Vertices in the graph, represent users and resources (e.g., pictures, posts, etc.) and edges of the graph are utilized to model the relationships among users and resources.

Figure 1 shows an example of such graph. In this example, there are three users: Alice, Bob and Carol and one resource, a picture, indicated as Pic. Dotted arrows represent social relationships between users, while plain arrows are relationships between users and resources. The `family` relationship between Alice and Bob is a bidirectional relation (*symmetric*), as in Facebook. On the other hand, `follow` is a unidirectional relation (*asymmetric*) which means that a user can follow others without being followed. Additionally, the plain arrows indicate the connections between users and the resource Pic. The connection between Alice and Pic denotes that Alice is the owner of the picture, while Bob is tagged in it.

2.1 Privacy Policies in Online Social Networks

Privacy settings from which hereafter we will refer to as *privacy policies*, in today OSNs allow users to set *who* can access *what* information. In most existing OSNs, users are provided with a large variety of relationships that they use to create their own social circles, e.g., family, friends, colleagues, hiking group, acquaintances and so forth. The current privacy policies in OSNs are specified in terms of relationships. Typically, granting access to a data item is subject to the type of the relationship or its composition (e.g., friends, friends-of-friends). Moreover, privacy policies depend on the functionality and purpose of the OSN platform. For instance, in privacy policies of Facebook, users can determine who can access their posts or friends lists and also, determine the actions that other users can perform. For example, users can define the social circle extent (e.g., friends-of-friends or everyone) of users who are able to send them a friendship request. Other OSNs, like Twitter, are very liberal with what they share to your news feed. Twitter users are able to choose whether to allow the general public to view their Tweets or to make them viewable only to people whom they approve to follow them. In private mode, users avoid having their Tweets viewed by anyone, and those Tweets will be out of any search engines such as Google. Unlike other OSNs, LinkedIn is based on full access and effortless disclosure of users professional and relevant information such as where they live, telephone numbers, occupations, professional connections and education. The profile acts as an online resume. However, since many users have privacy concerns, LinkedIn helps users to manage their privacy. A user can specify who can see her *connections list*, and controls whether or not other users can see if the user has viewed their profiles. Unlike other networks, LinkedIn allows users to control who can see their profile photos. For instance, "My Connections" means the profile photo will be seen by connection list only, while the "Network" option means that the user profile picture can be viewed by second and third level connections. Finally, in all OSNs, users can block other users. Blocking a user means she virtually does not exist anymore and prevents any interaction between the two users, hence excluding any abusive or harmful behaviour.

Although privacy policies enable users to protect their personal information from other users within the network, they are notoriously difficult to configure correctly and do not easily match the OSN's users sharing intentions [2, 4, 20, 38, 45, 46, 47, 48]. Madden conducted a study where she found 48% of OSN users report struggles in managing their OSN privacy settings [46]. Another study reveals that only 37% of Facebook's users consider that privacy settings match their expectations [45]. Due to this lack of understanding and management difficulties, some users leave their privacy settings to the *default*, which mostly provides insufficient protection, allowing their data be exposed to strangers [27, 43].

Although OSN companies are putting effort into improving privacy policies for their users, there are still many privacy risks for their users that are caused by limitations in their current privacy protection mechanisms [14, 52]. The research community has identified a number of desirable features, that are considered to be requirements, in order to enhance privacy in OSNs [31, 66]. Privacy policies should be:

- *Understandable and User-friendly*: OSN's users on average do not have expertise in choosing the proper privacy policies that can protect their information from undesired disclosure [49]. Users require privacy mechanisms that are easily understandable and manageable, and user friendly interfaces that help them to reduce the burden of setting their privacy policies.

- *Interoperable*: Usually, users have accounts on different OSNs such as Facebook, LinkedIn, Instagram, etc. since each OSN has a specific objective. For example, the purpose of Facebook is to facilitate users' contact in their social relationships, while LinkedIn allows users to maintain their professional network. Thus, it would be desirable for the privacy mechanisms and policies to be interoperable to protect cross-border data flows between OSNs.

- *Sticky*: Users' data can span several organizations, applications and locations. Therefore, policies have to be *sticky* which means they should follow the piece of data to describe how it can be used after access to it has been granted.

- *Fine-grained*: Providing fine-grained privacy policies will make it easier for users to express their privacy preferences. For instance, a user should be able to specify privacy policies for individual blog entries, a specific picture (or part of it), or comments. Such privacy policy formulation can impact users' understanding of privacy policies and thus their success in appropriately employing them. Although Facebook is a good example of how OSNs have come a long way in implementing fine-grained privacy policies, its privacy protection mechanism still lacks key elements. One of the most important is the absence of distinguishability between accessing and sharing policies.

- *Relationship-based*: People share their information with others based on the relationship they have with them. The properties of the relationship such as strength, direction, type, etc. also affect the way that people share and disclose their personal information. This is supported by many studies in social psychology [25, 35, 62]. Consequently, there is a wide agreement that privacy control in OSNs should be modeled based on relationships that arise as consequence of the structure of OSNs (e.g.,[16, 17, 30, 32]).

- *Specific to the type of content*: In OSNs, users are able to post and share different pieces of information as texts, photos, videos, comments, audio, events, hobbies, location, website (url), etc. A number of studies demonstrate that the format of the information and its content have impact when defining privacy policies, and that there is a relationship between content types and the way users communicate (like, share, comment) [15, 41, 42, 64]. For example, Kim and Yang [41] conducted a study where they found out that posts containing pictures or videos were more likely to motivate users to like and share them.

- *Multiparty*: In the definition of a privacy policy all users who are related to a given piece of information, and such that their privacy can be affected by that, should be involved in deciding who should access that item. Furthermore, privacy protection mechanisms have to be able to automatically detect which pieces of information are co-owned by whom. As the protection mechanism contains more than one privacy preference, most of the time users, considered here as *co-controllers*, have different desires regarding who they want to view this piece of information. So, it is possible that, within a privacy policy, a user is permitted and denied access at the same time, which creates a conflict in the policies. Suppose that Alice takes a picture of her and Bob, and uploads it to her space[1] and tags Bob. Assume that Alice sets that her friends can see her picture and that Bob wants to customize his privacy policies to include only his close friends. In this case Bob's privacy preference will conflict with some of Alice's friends. *Multiparty privacy* is designed to facilitate the harmonization of collectively held privacy policies by all users that co-control a piece of information, where their privacy might be lost depending on with whom the co-controlled piece of information is shared.

- *Trust-based*: Besides being relationship-based, privacy policies need to take into account a value for the trust of the relationships. Trust is an important component for building relations in online communities where users can post and share their personal information, experiences, social activities and opinions without concerns about privacy. Many studies corroborated that trust has a significant effect on the level of information disclosure between users [39]. Thus, in order to balance the open nature of OSNs and preserve the privacy concerns of

[1] In this work, we use the term *space* to refer both to the user's profile and her interactive arena. A user *profile* is a collection of settings and information associated with the user. It may be defined to be the explicit digital representation of the user's identity in the context of the given (OSN) environment. It includes information such as age, location, and interests. The user *interactive arena* is the arena for both public interaction and communication with others (e.g., the wall in Facebook, the Home timeline in Twitter, stream in Diaspora, etc.).

users, trust has to become a critical factor in privacy protection mechanisms.

In order to develop a successful privacy protection mechanism for OSNs, all of the above features are desirable. In this thesis, we focus only on *Multiparty* and *Fine-grained* privacy policy features. In what follows, we discuss and give examples of problems arising from the lack of having these features.

Multiparty privacy policy

As we mentioned, users can control access to their data by using the privacy settings that OSNs provide. They are implemented based on relationships between the owner of the data item and other users. All the data items in the user space are owned by that user, so we say that the user is the *owner* of such data items. The relationship between the data item owner and other users depends on the OSN platform and the data item owner's preference, e.g., friends, followers, friends of friends, etc. The privacy protection mechanism implemented in current OSNs allows users to manage the access to data items that are uploaded by them or posted in their space. Besides uploading and posting, users engage in communication on OSNs via behaviors that are dependent on the structure of the OSN's platform and its purpose. For instance, Facebook offers the following five behaviors: *like, comment, tag, mention and share*[2]. Whenever a data item is shared, or has tagged or mentioned users, all the users who are involved should be able to express their privacy policies. Consider the following example. Assume that Alice uploaded a picture of herself drinking with her friend Bob. In this case, the picture is doubtlessly co-owned by both Alice and Bob. In Alice's privacy setting, she sets that this picture can be viewed by her friends, colleagues and family. Bob considers this picture to be sensitive and does not want to share it with Alice's family and colleagues. Thus, Bob's privacy will be violated as his picture will be viewed by Alice's family and friends. In order to protect him from such privacy violation, the OSN privacy protection mechanism has to be properly designed, considering and respecting the privacy preferences of all involved users.

Hundreds of billions of data items that are uploaded and shared in OSNs are co-owned by more than one user [37, 63]. Enge conducted a study on 4 million tweets and found that tweets with pictures acquired more than double the retweets and likes than a pictureless tweet would [26]. A study procured by Mention's Twitter Report [50], showed that almost 40% of tweets

[2] In the thesis, we consider tagging, mentioning and sharing behaviors.

include the mention symbol to engage more people in the conversation of the tweet and alert others to subjects of interest. Scholars have emphasized that tagging is one of the popular behaviours on OSNs (e.g., Facebook), where users commonly tag and are tagged [9, 22, 23]. Nowadays on most OSNs it is possible only for the owner of the space to specify the privacy policies of co-owned data items regardless of the privacy preferences of other users who are identified in these data items. Current OSNs offer limited support for managing co-owned data items where users can only use strategies like untagging or reporting inappropriate content.

When OSNs' tags are normally used, users upload a picture and name other users in it with a link to their profile. There are some OSNs in which the tagged users receive notifications about the pictures they have been tagged in in order to approve them before the picture appears on their space. If the tagged users in a picture do not want to share it with their list of connections (e.g., the user's Friends List in Facebook), they can untag themselves from it. However, this strategy does not fulfill the users desires for different reasons. Firstly, when tagged users untag themselves from a picture, it does not mean that the actual picture is removed, or that they block the possibility of the picture to be accessed by undesired users. For example, assume that Alice and Bob are in a picture taken during their holiday trip. Alice decided to upload it to Facebook (making her the owner) tag Bob and share it with her friends. When Bob received the tag notification, he decided not to share the picture with his friends list; thus, he took action not to make the picture appear on his Facebook Timeline by not approving the tag (untagged himself). Despite Bob untagging himself and the picture now not appearing in his own Timeline, the picture will still be viewed by other users according to Alice settings. So if Bob has common friends with Alice, all these friends will be able to view the picture in Alice's Timeline because they are her friends too. This means, the act of tagging brought Bob into a state where he became connected to the picture by the choice of Alice, and despite untagging himself, he is still connected to the picture elsewhere in the network (Alice's space) and accessing and visualizing this tagged picture only depends on Alice's privacy policies.

For Bob, the fact that he is related (as co-owner) to the picture does not give him any control over who can access it from his social circle (like Friends List in Facebook) or the general social network. This lack of control stems from the fact that Alice has full rights to determine who can access the picture. The second limitation in the untagging strategy is that the tagging is limited to one type of an OSN's data item, a picture, but is not available for other items such as posts, events, comments. In posts and comments, OSNs users can interact with each other through a mention feature that creates a direct link to the mentioned user's profile. In some OSNs (like Facebook and Twitter) the mentioned users receive a notification that they were mentioned. However, these posts or comments that have the mentions are solely controlled by users who created them. Finally, even though untagging is used as a management strategy [9], users expressed occasional discomfort at offending the user who tagged them in the picture by using the untagging feature [7, 8]. In some extreme situations tagged users are forced to ask the uploader personally to remove the data item or remove the relationship links.

Most OSNs provide an option for users to report and make a request to eliminate a data item published by others, for example, because it is inappropriate. This option does not guarantee that the data item will definitely be eliminated, and it is mostly used to handle highly inappropriate contents such as harassment, hate speech, child endangerment, violence, nudity or others which are considered publicly offensive. Hence, this mechanism does not solve the privacy issue due to co-ownership. In many scenarios the privacy violations do not necessary happen within the context of an offence. For instance, tagged/mentioned users may simply not feel open to share some information with other users due to privacy concerns, or they may want to restrict the view of this information to a smaller or specific audience. Also, it is important to point out that reporting is not a proactive mechanism because flagging the data item as inappropriate normally happens after uploading. So, reporting may be too late, the privacy breach may have already occurred and the damage may have already been caused, or users may have been able to copy the data item and disseminate it using other platforms. The bottom line is that compensatory solutions are in general not enough [58, 59].

Coarse-grained privacy policy

One of OSNs shortcomings is the inflexibility of privacy policies to accommodate the user's needs and intentions. This lack of flexibility makes many privacy protection features, still in demand by users, unfulfilled whilst a few privacy violations from other users remain unsolved. In practice, many OSNs (such as Facebook) have already provided helpful and desirable privacy policies, some of which can help users to manage their data items. For instance, in Facebook users can specify policies like "Only my friends can see my friends list", "Only friends of my friends can send me friends requests", "Only my friends can see what others post on my Timeline" or "The picture where I was tagged should not appear on my Timeline unless I approve it". Moreover, Facebook's users can share their data items with a wide range of predefined users, including friends, groups, friends of friends, or all [13]. These features enable users (the owner) to define which data items can be viewed and by which users. However, given their importance for users, many policies are missing in current privacy protection mechanisms. In Facebook, users can not state policies like "Only the users who live in the same country as me can send me a friend request", "I do not want to be tagged in pictures by anyone other than the members of my close friends group", "Nobody apart from my family group can know my child's location" or "My post can be seen by my friends and friends of friends but nobody apart from my family group can share it".

Also, current OSNs' privacy protection mechanisms offer the same option of privacy policies without given consideration to the type of data item to be protected, whether it be a video, a picture, a location, an event, etc. Thus, users cannot specify their privacy policies according to the type of data item; for instance, in Facebook users cannot choose a policy like "Only my friends can see a post having my location", "Only my close friends group can see a post with a video". Furthermore, presently the privacy protection mechanisms do not equip the users with feature(s) to identify the level of privacy concern(s) that they have with regards to their data item. For example, users are not able to express policies like "I have high sensitivity level for all posts containing location" or "I have medium sensitivity level for all pictures that I have been tagged in".

As we mentioned, current privacy protection mechanisms are built based on relationships. By their very nature, each relationship has its own intensity (degree of trust), thus relationships among users are not equal [34]. However, the privacy protection mechanisms make no effort to distinguish between users' relationships; users are either in a relationship or are strangers. For example, in Facebook, users are classified either as friends or outsiders with nothing to choose from inbetween. At present, privacy protection mechanisms ignore the existence of trust differences; thus, users are neither able to express how much they trust other users nor to state policies by using a trust concept. Again, using Facebook as an example, users cannot specify policies that apply trust like "Whenever I am tagged by my friends whom I highly trust, the picture can be shown on my Timeline without approval" or "My posts can only be shared by highly trusted friends".

In general, OSNs' privacy policies determine who can access which data items along with other special privacy policies, which differ from one OSN to another according to their characteristics and functionality (e.g., untag in Facebook). However, looking deeper at what users can do with someone else' data items, there are two main actions: accessing and disseminating (sharing) the data items. Since most privacy policies are centered around the data items accessibility that sharing policies are embedded within it, it can be inferred that privacy policies are indirectly in control of sharing, although the two actions are functionally different. Using Facebook as an example, suppose that Alice posted a picture specifying friends of friends privacy policy. This means it can be seen by her friends and their friends. Bob who is her friend can not only see the picture but even share it with his friends, friends of friends or everyone. Imagine now that Bob decides to share it with his friends of friends; this action will increase the picture's audience to Alice's friends of friends and Bob's friends of friends. Despite that Alice set her privacy policy for only her friends of friends to see it, the picture is shared by her friend Bob and shown to more people than she expected. This lack of having policy options about sharing leads to undesirable results and privacy breaches.

In view of this, privacy policies need to be flexible to accommodate the user's needs and intentions, and more fine-grained settings are needed. However, fine granularity and flexible privacy policies may lead to an overwhelming and complex cognitive demand for the users to deal with them. This represents a burden that could worsen user's tendency to ignore policy specification, and trust the default privacy policies. This obeys to the OSNs' interests rather than the users. Additionally, such restricted privacy policies reduce the amount of data items shared in the OSNs; it makes the OSN platforms less likely to attract more users, which inevitably decreases their growth. So, an equilibrium between too little flexibility and an excessively complicated privacy policy management is needed.

2.2 Access Control in Online Social Networks

The privacy protection mechanism that is employed in OSNs today to control the access and dissemination of users' information is *access control* [1]. Access control mechanisms regulate how a subject may access an object [33] and is one of the most important features of today's systems to protect access to data items [6]. It has three main concepts: setting the policies that authorize certain individuals to access certain data items; authenticating evidence associated with an access request; assessing the access request based on the given policies [33].

The access control model behind the privacy protection mechanism of OSNs is readily distinguishable from others access control mechanisms, such as Mandatory Access Control (MAC) [53], Discretionary Access Control (DAC) [55], Attribute-Based Access Control (ABAC) [65] and Role-Based Access Control (RBAC) [54]. Several access control models that have been developed in recent years are aimed at effectively capturing the nature of information accessed and shared in OSNs [12, 17, 18, 19, 29, 30]. Several studies showed ample evidence that users' relationships should be considered as a central concept in modeling the privacy protection mechanism of OSN [17, 29, 31, 35, 62]. In what follows, we provide background on the access control model OSNs implement.

Relationship-Based Access Control

In *Relationship-Based Access Control* (ReBAC), authorizations are specifically based on relationships between users. This new access control paradigm was initially inspired by the structure of OSNs. Comparing with other access control models such as DAC, RBAC or ABAC, there are three distinguishing features of ReBAC model that are identified by Fong et al. [29].

- *ReBAC model relies on leveraging social relationships as access policies.* So, users specify the audience of their data items based on their direct or indirect relationship with others. Consider Alice, in Figure 1, who may state a policy that makes Pic solely accessible to her family members, that is Bob. Alternatively, she can specify that Pic can be accessed by her indirect family members (i.e., family of family) which results in Bob's family members. Another policy option can be public which gives accessibility to everyone in OSN's platform.

- *Before granting the access to any data item, it has to be reachable in the social graph.* Reachability is a necessary step prior setting policies regarding accessibility. In Figure 1 for example, Bob is reachable from Alice through *family* relation. Thence, Bob can access Pic if permission is granted. On the other hand, Carol would not be able to access Pic, since Alice and Bob are not reachable by him.

- *Abstraction of the succession of events and interactions that take place in the system.* According to many access control systems, authorization is a function that depends on the sequence of events and interactions [56]. As shown in Figure 1, family relationship is a bidirectional relation that indicates to both users that they are required to agree before establishing the relationship. Envision that Carol wants to access Alice's picture (Pic). In order to achieve that one possible option can be as following: 1) Carol sends a friendship request to Alice; 2) Alice accepts the request; 3) Carol would be able access the Pic if Alice sets policies that grant her friends accessibility. In ReBAC, the sequence of events is abstracted into the social graph. Continuing with our previous example, the impact of Alice and Carol becoming friends produces a friend relationship between them. This abstraction of the event history becomes a basis of authorization decisions in ReBAC.

ReBAC is a paradigm that captures the nature of information accessing in OSNs by taking into account users' relationships as a core concept [30]. However, the ongoing privacy violations in OSNs indicate that ReBAC as applied on OSNs has limitations which means this model might need to be revised. Mondal et al. discuss insufficiency of managing privacy in OSNs by applying access control, while Fogues et al. discuss a few open challenges in ReBAC for OSNs [28, 51]. They bring up, among others, the following issues:

1. *A privacy protection mechanism is needed to enforce the privacy preferences of all involved users when dealing with a data item that is related to other users.* This issue relates to the aforementioned problem of lack of having *Multiparty* policies. To address multiple ownership in ReBAC, the model has to be extended to consider all different relationship contexts between a co-owned data item and the involved users. Moreover, the model should focus on supporting the detection and resolution of multiparty privacy conflicts, since individual privacy preferences may conflict because other involved users in the co-owned data item may want to grant access to or share it with different audiences. The recent line of work on privacy management and access control for OSNs is tackling this issue [58].

2. *Privacy policies that OSNs provide do not capture how data items should be disseminated.* In OSNs, it is also important for the owner of the data item or all involved users in case of co-owned data to specify privacy policies that limit who can disseminate their data items. For example, Alice might feel open to permit Bob an access to Pic, but maybe she does not want him to share it with Carol who, originally, did not have the right to view the Pic. It is important to design a mechanism that empowers users to express privacy policies that determinate who can disseminate their data items and who cannot. Ongoing work on controlling a shared data item is handling this issue [32, 44]. This issue is related to the problem arising from lack of *Fine-grained* policy.

All in all, these two privacy issues show that OSNs' privacy protection mechanisms should be supportive by multiparty privacy management as traditional single-user approaches lack the flexibility to accommodate the co-owned data scenarios, causing undesired disclosure of sensitive data. Moreover, OSNs' privacy policies should be more expressive. In the next section, we introduce our approach to address some of the privacy issues we have described.

3 Thesis Overview

Our aim in this work is to provide a framework that empowers OSNs' users to collectively manage viewing and sharing their co-owned data items. As conflicting policies are commonly raised in multiple ownership privacy protection mechanisms, we proposed *Viewing* and *Sharing* aggregation-based algorithms which make a decision by solving potential conflicts between the different privacy settings of all the concerned users. This is achieved by taking into account the following aspects: the trust among users; the sensitivity level of users with respect to the concerned data item; and the weights of the following: (i) the types of *controllers* (those who are concerned in the decision that determines who can access a given data item and who cannot) and (ii) the types of *accessors* (those who are identified to access a given data item or not). We evaluated our solution by generating all possible combinations of components and performed experiments to show how the different components affect the decision on who should or should not, access or share the data items. Furthermore, we provided proof-of-concept implementation into the open source OSN Diaspora. This work is currently under submission to the Journal of Logical and Algebraic Methods in Programming (JLAMP) 2019 and was co-authored by Raúl Pardo, Gerardo Schneider and Pablo Picazo-Sanchez.

Regarding my contributions, I proposed the collaborative access control model, formalized the policies and developed the collaborative access control algorithms. Moreover, I implemented the proof-of-concept prototype in Diaspora.

Bibliography

[1] S. Abiteboul, R. Agrawal, P. Bernstein, M. Carey, S. Ceri, B. Croft, D. DeWitt, M. Franklin, H. G. Molina, D. Gawlick, et al. The lowell database research self-assessment. *Communications of the ACM*, 48(5):111–118, 2005.

[2] A. Acquisti and R. Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *International workshop on privacy enhancing technologies*, pages 36–58. Springer, 2006.

[3] I. Altman. A conceptual analysis. *Environment and behavior*, 8(1):7–29, 1976.

[4] J. Anderson and F. Stajano. Must social networking conflict with privacy? *IEEE Security & Privacy*, 11(3):51–60, 2013.

[5] U. G. Assembly. Universal declaration of human rights. *UN General Assembly*, 1948.

[6] E. Bertino and R. Sandhu. Database security-concepts, approaches, and challenges. *IEEE Transactions on Dependable and secure computing*, (1):2–19, 2005.

[7] A. Besmer and H. Lipford. Tagged photos: concerns, perceptions, and protections. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4585–4590. ACM, 2009.

[8] A. Besmer and H. Richter Lipford. Moving beyond untagging: photo privacy in a tagged world. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1563–1572. ACM, 2010.

[9] J. Birnholtz, M. Burke, and A. Steele. Untagging on social media: Who untags, what do they untag, and why? *Computers in Human Behavior*, 69:166–173, 2017.

[10] D. M. Boyd and N. B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of computer-mediated Communication*, 13(1):210–230, 2007.

[11] N. Bronson, Z. Amsden, G. Cabrera, P. Chakka, P. Dimov, H. Ding, J. Ferris, A. Giardullo, S. Kulkarni, H. Li, et al. {TAO}: FacebookâĂŹs distributed data store for the social graph. In *Presented as part of the 2013 {USENIX} Annual Technical Conference ({USENIX}{ATC} 13)*, pages 49–60, 2013.

[12] B. Carminati, E. Ferrari, and A. Perego. Rule-based access control for social networks. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1734–1744. Springer, 2006.

[13] F. H. Center. What are the privacy settings for groups?, 2019. `https://www.facebook.com/help/220336891328465#What-are-the-privacy-options-for-groups`, Last accessed on 2019-3-23.

[14] T. H. Center. How to protect and unprotect your tweets, 2019. `https://help.twitter.com/en/safety-and-security/how-to-make-twitter-private-and-public`, Last accessed on 2019-3-18.

[15] K. Chauhan and A. Pillai. Role of content strategy in social media brand communities: a case of higher education institutes in india. *Journal of Product & Brand Management*, 22(1):40–51, 2013.

[16] Y. Cheng, J. Park, and R. Sandhu. Relationship-based access control for online social networks: Beyond user-to-user relationships. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 646–655. IEEE, 2012.

[17] Y. Cheng, J. Park, and R. Sandhu. A user-to-user relationship-based access control model for online social networks. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 8–24. Springer, 2012.

[18] M. Cramer, J. Pang, and Y. Zhang. A logical approach to restricting access in online social networks. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies*, pages 75–86. ACM, 2015.

[19] J. Crampton and J. Sellwood. Path conditions and principal matching: a new approach to access control. In *Proceedings of the 19th ACM symposium on Access control models and technologies*, pages 187–198. ACM, 2014.

[20] B. Debatin, J. P. Lovejoy, A.-K. Horn, and B. N. Hughes. Facebook and online privacy: Attitudes, behaviors, and unintended consequences. *Journal of computer-mediated communication*, 15(1):83–108, 2009.

[21] J. W. DeCew. *In pursuit of privacy: Law, ethics, and the rise of technology.* Cornell University Press, 1997.

[22] A. Dhir, G. M. Chen, and S. Chen. Why do we tag photographs on facebook? proposing a new gratifications scale. *new media & society*, 19(4):502–521, 2017.

[23] A. Dhir et al. Exploring online self-presentation in computer-mediated environments-motives and reasons for photo-tagging and untagging. 2016.

[24] E. Directive. 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal of the EC*, 23(6), 1995.

[25] S. Duck. *Human relationships.* Sage, 2007.

[26] E. Eric. Twitter engagement unmasked: A study of more than 4m tweets, 2014. `https://www.stonetemple.com/twitter-engagement-umasked/`, Last accessed on 2019-3-20.

[27] M. Fire, D. Kagan, A. Elyashar, and Y. Elovici. Friend or foe? fake profile identification in online social networks. *Social Network Analysis and Mining*, 4(1):194, 2014.

[28] R. Fogues, J. M. Such, A. Espinosa, and A. Garcia-Fornes. Open challenges in relationship-based privacy mechanisms for social network services. *International Journal of Human-Computer Interaction*, 31(5):350–370, 2015.

[29] P. W. Fong, M. Anwar, and Z. Zhao. A privacy preservation model for facebook-style social network systems. In *European Symposium on Research in Computer Security*, pages 303–320. Springer, 2009.

[30] P. W. Fong and I. Siahaan. Relationship-based access control policies and their policy languages. In *Proceedings of the 16th ACM symposium on Access control models and technologies*, pages 51–60. ACM, 2011.

[31] C. Gates. Access control requirements for web 2.0 security and privacy. *IEEE Web*, 2(0), 2007.

[32] R. Gay, J. Hu, H. Mantel, and S. Mazaheri. Relationship-based access control for resharing in decentralized online social networks. In *FPS*, pages 18–34, 2017.

[33] D. Gollmann. Computer security. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):544–554, 2010.

[34] M. S. Granovetter. The strength of weak ties. In *Social networks*, pages 347–367. Elsevier, 1977.

[35] D. J. Houghton and A. N. Joinson. Privacy, social network sites, and social relations. *Journal of Technology in Human Services*, 28(1-2):74–94, 2010.

[36] B. Howard. Analyzing online social networks. *Commun. ACM*, 51(11):14–16, Nov. 2008.

[37] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis. Face/off: Preventing privacy leakage from photos in social networks. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 781–792. ACM, 2015.

[38] M. Johnson, S. Egelman, and S. M. Bellovin. Facebook and privacy: it's complicated. In *Proceedings of the eighth symposium on usable privacy and security*, page 9. ACM, 2012.

[39] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

[40] S. Joseph and M. Castan. *The international covenant on civil and political rights: cases, materials, and commentary.* Oxford University Press, 2013.

[41] C. Kim and S.-U. Yang. Like, comment, and share on facebook: How each behavior differs from the other. *Public Relations Review*, 43(2):441–449, 2017.

[42] P. Klemperer, Y. Liang, M. Mazurek, M. Sleeper, B. Ur, L. Bauer, L. F. Cranor, N. Gupta, and M. Reiter. Tag, you can see it!: using tags for access control in photo sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 377–386. ACM, 2012.

[43] B. Krishnamurthy and C. E. Wills. On the leakage of personally identifiable information via online social networks. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 7–12. ACM, 2009.

[44] A. Lazouski, F. Martinelli, and P. Mori. Usage control in computer security: A survey. *Computer Science Review*, 4(2):81–99, 2010.

[45] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 61–70. ACM, 2011.

[46] M. Madden. Privacy management on social media sites. *Pew Internet Report*, pages 1–20, 2012.

[47] M. Madejski, M. Johnson, and S. M. Bellovin. A study of privacy settings errors in an online social network. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 340–345. IEEE, 2012.

[48] M. Madejski, M. L. Johnson, and S. M. Bellovin. The failure of online social network privacy settings. 2011.

[49] A. Mazzia, K. LeFevre, and E. Adar. The pviz comprehension tool for social network privacy settings. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 13. ACM, 2012.

[50] Mention. The twitter engagement report 2018,users tagged, 2018. `https://mention.com/en/reports/twitter/users-tagged/`, Last accessed on 2019-3-20.

[51] M. Mondal, P. Druschel, K. P. Gummadi, and A. Mislove. Beyond access control: Managing online privacy via exposure. In *Proceedings of the Workshop on Useable Security*, pages 1–6, 2014.

[52] F. newsroom. Hard questions: What is facebook doing to address the challenges it faces?, 2019. `https://newsroom.fb.com/news/2019/02/addressing-challenges/`, Last accessed on 2019-3-18.

[53] R. S. Sandhu. Lattice-based access control models. *Computer*, 26(11):9–19, 1993.

[54] R. S. Sandhu. Role-based access control. In *Advances in computers*, volume 46, pages 237–286. Elsevier, 1998.

[55] R. S. Sandhu and P. Samarati. Access control: principle and practice. *IEEE communications magazine*, 32(9):40–48, 1994.

[56] F. B. Schneider. Enforceable security policies. *ACM Trans. Inf. Syst. Secur.*, 3(1):30–50, Feb. 2000.

[57] D. J. Solove. *Understanding privacy*, volume 173. Harvard university press Cambridge, MA, 2008.

[58] J. M. Such and N. Criado. Multiparty privacy in social media. *Commun. ACM*, 61(8):74–81, 2018.

[59] J. M. Such, J. Porter, S. Preibusch, and A. Joinson. Photo privacy conflicts in social media: A large-scale empirical study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3821–3832. ACM, 2017.

[60] S. D. Warren and L. D. Brandeis. Right to privacy. *Harv. L. Rev.*, 4:193, 1890.

[61] A. F. Westin and O. M. Ruebhausen. *Privacy and freedom*, volume 1. Atheneum New York, 1967.

[62] E. Wiese, A. Wykowska, J. Zwickel, and H. J. Müller. I see what you mean: how attentional selection is shaped by ascribing intentions to others. *PloS one*, 7(9):e45391, 2012.

[63] H. Xu. Reframing privacy 2.0 in online social network. *U. Pa. J. Const. L.*, 14:1077, 2011.

[64] C.-m. A. Yeung, L. Kagal, N. Gibbins, and N. Shadbolt. Providing access control to online photo albums based on tags and linked data. In *AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0*, pages 9–14, 2009.

[65] E. Yuan and J. Tong. Attributed based access control (abac) for web services. In *IEEE International Conference on Web Services (ICWS'05)*. IEEE, 2005.

[66] C. Zhang, J. Sun, X. Zhu, and Y. Fang. Privacy and security for online social networks: challenges and opportunities. *IEEE network*, 24(4):13–18, 2010.

1

A Collaborative Access Control Framework for Online Social Networks

Hanaa Alshareef, Raúl Pardo, Gerardo Schneider, Pablo Picazo-Sanchez

Journal of Logical and Algebraic Methods in Programming 2019

Abstract. Most Online Social Networks allow users to set their privacy settings concerning posting information, but current implementations do not allow a fine grained enforcement in case the posted item concerns other users. In this paper we propose a new collaborative access control framework that takes into account the relation of multiple users for viewing as well as for sharing items, eventually solving conflicts in the privacy settings of the users involved. Our solution is based on the sensitive level of users with regard to the posted item and on trust among users. We provide a thorough evaluation of our framework where we focus on how varying some of the parameters directly influence the outcome of the permitting/denying decision of the proposed algorithms. Last but not least, we present a proof-of-concept implementation of our approach in Diaspora, an open source social network.

1 Introduction

Most Online Social Networks (OSNs) today have privacy settings that allow users to define their preferences in what concerns the use of their data. This usually includes aspects related to whom can have access to which information but it is limited in a number of ways. For example, in OSNs like Facebook or Twitter users may only describe who is the direct audience of a given item (post, message, picture, etc.), meaning that it only concerns who has access to the item based on the explicit relationships the user has previously defined. In many cases it involves only one level in the relationship order or two levels, e.g., friends or friends of friends. This is a limitation since users might be interested in defining privacy policies that limit the access to other users not directly connected with them beyond two levels. This is the case, for instance, whenever somebody who originally got access to the information, wants to share it with other users unrelated to the original source of the item.

In the majority of OSNs, the audience of a piece of information uploaded to the system is solely defined by a single user. Typically, the user defining the audience is the one uploading the data, be to her own *space*, or somewhere else.[1] However, many other users may also be concerned with the posted data, so they should also have a say in who may access or not. Ideally, there should be a mechanism allowing all the involved users to take a decision collaboratively.

Current implementations of social networks rely on the so called Relationship-based Access Control (ReBAC) model [12] where the social relationships between users are used to express access control policies. Though ReBAC has been shown to have many advantages with respect to other access control models in OSNs [10, 12] it does not allow a fine grained enforcement in case

[1] In this work, we use the term *space* to refer both to the user's profile and her interactive arena. A user *profile* is a collection of settings and information associated with the user. It may be defined to be the explicit digital representation of the user's identity in the context of the given (OSN) environment. It includes information such as age, location, and interests. The user *interactive arena* is the arena for both public interaction and communication with others (e.g., the wall in Facebook, the Home timeline in Twitter, stream in Diaspora, etc.).

the posted item concerns many users, and the privacy settings usually do not allow for setting limits when a user wants to share the item she got access to. This lack of collaborative policies for access control may violate the privacy of the users who are part of the uploaded content, since they cannot decide who should access it: only the uploader of the data can decide that.

Additionally, apart from the aforementioned problem, ReBAC does not properly address users' policy conflicts. It is possible that, within an access control policy, a user is permitted and denied access at the same time, thus creating a conflict in the policy. Thus, there is a need to solve the conflicts before deciding who has access to the shared object [29, 30, 31].

A promising line of work for collaborative access control is the so-called *aggregation-based models* [29]. Using this approach the individual privacy preferences of all users related to an item are aggregated to decide, for instance, whether the item can be shared. However, the main drawback is that existing models (such as [17, 35]) are too coarse grained to cover all cases, and, in some cases, rely again on the data owner to choose a conflict resolution strategy.

We propose a *Viewing* and a *Sharing* aggregation-based algorithms which take a decision by solving potential conflicts between the different privacy settings of all the concerned users. Our algorithms rely on four different components: the sensitivity level of the users with respect to the concerned item, a trust relationship between users, and different weights for both the *controller types* and *accessor types*.[2] In order to be as general as possible, trying to cover most of the existing OSNs nowadays, we include in our model *factors* to give (or take) importance to some components. In this way, by giving different values to such factors we can obtain different results in what concerns the decisions to grant or deny viewing/sharing capabilities. This gives us the possibility to tune the decision policy, getting the outcome to range from very conservative (e.g., a strong denial from one party may overrule all the others) to more liberal (e.g., a majority granting access impose their decision).

We evaluate our solution by generating all possible combinations of the components under consideration (for a given value of the factors), and we perform experiments to show how the factors influence the decision on who should, or should not, access or share the posted items. Additionally, we provide a proof-of-concept implementation into the open source OSN Diaspora [8].

In summary, our **contributions** are:

[2] *Controller* and *accessor types* will be defined in Section 2; for the time being it suffices to know that they represent all the different users concerned with the item under consideration.

- A collaborative access control framework for OSNs taking into account: a trust relationship between users, sensitivity level of the users with respect to the concerned item, and different weights for both the controller types and accessor types (Section 2);
- An algorithm for collaboratively deciding who has access to an item (Section 2.3.1), and an algorithm to take such decision in case of sharing the item (Section 2.3.2);
- An evaluation of the behaviour of our algorithms based on an analysis of how the different components affect the decision to grant or deny access and sharing (Section 3);
- A proof of concept implementation of our framework in Diaspora (Section 3.1).

We compare our approach with previous work in Section 4, and we conclude in the last section.

2 A Collaborative Access Control Framework for OSNs

Our framework consists of three components: 1) an OSN model (Section 2.1); 2) access control policies (Section 2.2), and; 3) the collaborative access control mechanism (Section 2.3).

2.1 OSN Model

OSNs are typically structured as graphs, where vertices represent users and items whereas the edges of the graph represent connections between nodes. Concretely, vertices in the model are split into *actors*, *items*, and *groups*. Actors represent the real users of OSNs.[3] Each actor has a *space*, which includes the user's profile and interactive arena. An *item* is a digital representation of the physical object (e.g.,picture, text) to be posted, shared, etc. A *group* represents a spot that connects a collection of users who have the same beliefs, interests, behaviours, etc.

We also consider *Relationship types* to represent connections between vertices in the graph. For simplicity of presentation, and without loss of generality, in the rest of the paper we assume that there can exist only one relationship between any two vertices in the graph. In what follows we formally describe the OSN model that we use thorough this paper.

[3] For us it is not important whether the user is a physical individual, or an institution or corporation.

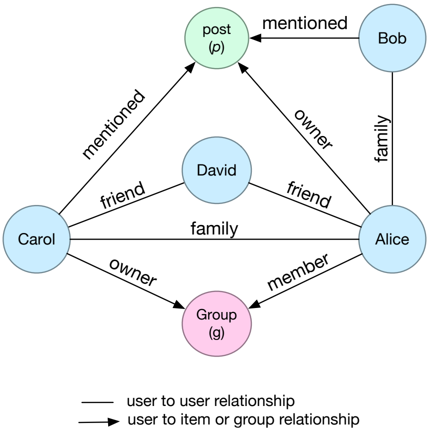


Figure 1.1: A Sample OSN Model

Definition 1 (OSN Model)**.** *Let $\mathcal{A}$ be a set of actors, $\mathcal{I}$ a set of items and $\mathcal{G}$ a set of groups. Consider also a set of relationship types $\mathcal{RT}$. An* OSN model *is a graph $SG = (\mathcal{A} \cup \mathcal{I} \cup \mathcal{G}, \{\mathcal{R}_i\}_{i \in \mathcal{RT}})$ where the vertices in the graph are elements of one of the sets $\mathcal{A}$, $\mathcal{I}$ or $\mathcal{G}$, and each $\mathcal{R}_i \subseteq (\mathcal{A} \cup \mathcal{I} \cup \mathcal{G}) \times (\mathcal{A} \cup \mathcal{I} \cup \mathcal{G})$ is a binary relation representing the edges of the graph.*

Figure 1.1 shows an example of an OSN. In this example, there are four actors, a post (p) and a group (g). The post is uploaded by Alice who *mentions* her family members Bob and Carol. Also, the group g was created by Carol, and Alice is a member of it. As it can be seen, the social network has relationships between actors and the item, e.g., *owner* (Alice *owns* the post p) and *mentioned* (Bob and Carol are mentioned in the post p); relationships between actors and groups, e.g., *owner* (Carol *owns* the group g) and *member* (Alice is a member of the group g); and relationships between actors[4], e.g., *family* (both Bob and Carol are in a family relationship with Alice) and *friend* (David is a friend of both Carol and Alice).

Trust Model. Trust becomes a crucial concept in OSNs for improving privacy mechanisms and reducing concerns about disclosing personal information [19]. Different techniques have been proposed in the literature (e.g., [4, 14, 20, 21]) to determine the optimum path and the trust value among users in OSNs. Ordinarily, people would like to express their trust using natural languages instead of numerical values. The *FuzzyTrust* algorithm [21]

[4] Here we consider that the relationships between actors are *symmetric*, though this might not be the case in general.

Table 1.1: Trust level weights

Linguistic term	Numerical value
none	0
low	0.25
medium	0.50
high	0.75
highest	1

allows us to do that, so in this work we use such algorithm to compute the trust between users. FuzzyTrust requires a *trust graph* where edges are labelled with the following set of *Trust Linguistic Terms* ($\mathcal{TLT}$) $\mathcal{TLT} = \{none, low, medium, high, highest\}$.

We assign to each of the elements of $\mathcal{TLT}$ a numerical value (see Table 1.1), used in our algorithms (Section 2.3). Hence, we define a trust graph as *TG*= ($\mathcal{A}$,$\mathcal{TR}$) where vertices are actors ($\mathcal{A}$), which represent the OSN users, and edges $\mathcal{TR} \subseteq \mathcal{A} \times \mathcal{TLT} \times \mathcal{A}$ are defined as a set of triples (a, tv, v), indicating that a's trust level for v is equal to tv. The function $infer : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ computes the numerical trust value between two actors in a trust graph (note that $(1 - infer)$ corresponds to the *distrust value*). Given a trust graph *TG*, we use the notation *TG.infer* to retrieve trust values among the actors in *TG*. When actors are directly connected in the trust graph, *infer* simply returns the numerical value corresponding to the label between the actors. On the contrary, if actors are not directly connected, *infer* applies the FuzzyTrust algorithm (see [21] for details). For instance, Alice may assign a high trust level for all her friends and a medium trust level to all her family except Bob who is assigned to a low level. The *infer* function computes the numerical value for trust between Alice and each of her friends, her family and Bob as following: 0.75, 0.50 and 0.25,respectively.

Associated Controllers. Similarly to [17], for each item we consider a set of *associated controllers*, representing all the actors of the OSN concerned in the decision of who should have access to a given item. We define a set of *controllers types* $\mathcal{CT} \subseteq \mathcal{RT}$ as $\{owner, stakeholder, contributor, originator\}$. Though the elements in $\mathcal{RT}$ may differ depending on the concrete OSN, we require that the elements in $\mathcal{CT}$ are always included in $\mathcal{RT}$. We define the set of *associated controllers* to item i as $C_i = \{c \mid (c, i) \in \mathcal{R}_j \text{ where } j \in C\}$ described in what follows.

Owner. All the items in the actor space are *owned* by that actor. We say that the actor is the *owner* of all those items. In our model we include an owner relation between an actor and an item every time an item is posted in her space. We use the relationship *owner* to indicate when an actor is the owner of an item.

Stakeholder. A *stakeholder* is an actor who is tagged or mentioned in an item. We use the relationship *stakeholder* to indicate when an actor is a stakeholder of an item.

Contributor. A *contributor* is an actor who posts an item in a space different than hers, e.g., Alice posting in Bob's space, making Alice to be a contributor for that post and Bob is the owner. We use the relationship *contributor* to indicate when an actor is a contributor of an item.

Originator. An actor is considered to be an *originator* when an item is shared from her space—note that the owner is the only one who can become an originator. For instance, if Alice shares an item from Bob's space to Carol's space, then Bob is the originator of the item. We use the relationship *originator* to indicate when an actor is an originator of an item.

Note that for each item there is exactly one owner, at most one contributor, at most one originator, and zero or more stakeholders.

Controllers Types Weight. In our framework the associated controllers do not necessarily have the same importance. We use the principle that close people tend to be similar [2, 9], and thus we give the stakeholders the same importance as the owner for two reasons: 1) they are explicitly mentioned (or tagged) without their approval, and; 2) stakeholders who are conventionally related to the content of the item, by some way or another, becomes liable to the disclosure of their sensitive information.

Regarding the contributor and the originator controllers, and contrarily to other proposals (e.g., [18, 27, 28]), we get them involved into the collaborative decision. The main difference with respect to the owner and stakeholder(s) is that we take into account the distance between them and the owner in the OSN model, and we give them different values. The *distance* between the owner and each associated controller is defined as the shorter connection path length between them considering all the relationships that connect them.

We propose two algorithms, *Viewing* and *Sharing*, corresponding to the different *actions* that an actor (associated controller) may do concerning an

Table 1.2: Controllers types' weights. Column 1 represents the types of associated controllers; column 2 shows the algorithms for producing the collaborative decisions; column 3 is the minimum distance among all the social relationships between the owner and the associated controller; column 4 shows the weight we assign to the associated controllers

Controller type	Algorithm	Distance	Weight
Owner	Viewing and Sharing	-	1
Stakeholder	Viewing and Sharing	-	1
Contributor	Viewing and Sharing	1	0.50
Contributor	Viewing and Sharing	≥ 2	0.25
Originator	Viewing	1	0.50
Originator	Viewing	≥ 2	0.25
Originator	Sharing	-	$\begin{cases} 0.25 & \text{if } \mathit{TG.infer}(\mathit{originator}, \mathit{owner}) \geq 0.75 \\ 0.75 & \text{otherwise} \end{cases}$

item. The algorithms depend on several parameters, among others the *weight* of the associated controllers.

Each associated controller is weighted based on whether she is involved in the process of making the collaborative decision regarding viewing or sharing an item, and her distance from the owner. Table 1.2 shows how this weight is defined for each associated controller in both algorithms. Note that for the owner and stakeholder types, the weight is always one and it does not depend on the distance since both are equally involved in the process of making a collaborative decision. In the case of the contributor, she is weighted differently based on her distance from the owner in both algorithms (see rows 3 and 4).

For the originator, the weight is different for each algorithm: for Viewing the distance is calculated in the same way as for the contributor, whereas for Sharing the distance is not relevant as she is weighted based on the trust level between herself and the owner. The trust level is used to indicate how much influence an originator's opinion will have on the aggregated decision. In this case, if the originator highly trusts the owner (with a value ≥ 0.75) then the weight is only 0.25, otherwise it will be 0.75. In other words, when the originator highly trusts the owner, the task of deciding whether to share or not is delegated to the trustee.

In what follows, we use a function $wct : \mathcal{CT} \rightarrow \mathbb{R}$ to retrieve the weight of a controller type.

2.2 Access Control Policies

In this section, we introduce the access control policies that our collaborative algorithms use. Before explaining what an access control policy is and how it is represented, we introduce two concepts: accessor specifications and sensitivity levels of data items.

Accessor Specification. In our framework each associated controller can identify a set of actors who can access her data and who cannot, the so-called *accessors.* Associated controllers can specify their permitted and denied accessors using the following accessor types: *actor names*, *group names* and *relationship names.* Formally, we define the set of accessor types as $\mathcal{AT} = \{an, gn, rn\}$. Many OSNs allow users to specify who can access their information using these accessor types; they have also been used in other collaborative access control frameworks like in [17]. These accessor types help the controllers to customize their access control policies.

Definition 2 (Accessor specification)**.** *We define an accessor specification as pair* $(\mathcal{A} \cup \mathcal{G} \cup \mathcal{RT}) \times \mathcal{AT}$*. We use* $\mathcal{AS}$ *to denote the set of accessor specifcations.*

For example, the accessor specification $\langle Alice, an \rangle$ indicates that Alice is specified as actor, while $\langle friends, rn \rangle$ denotes that controller determines her friends relationship either to access her data or not. We denote the universe of accessor specifications as, $\mathcal{U}_{\mathcal{AS}} \subseteq 2^{\mathcal{AS}}$. Accessor types are organized hierarchically forming a total order: $rn > gn > an$ where $y > x$ means that x is *more specific than* y (or that y is *more general than* x). As we define below, denying or permitting an actor by means of a policy that uses a specific accessor type contributes more to the final decision than policies using less specific accessor types.

Accessor Type Weight. In our framework, not all accessor types are equal. For example, we consider that directly denying an actor ("Alice is denied") should have a "stronger" effect on a collaborative decision than indirectly denying an actor because it belongs to a relationship ("My friends are denied", where Alice is one of the friends). Thus, we weight the accessor types based on the *most-specific-takes-precedence* principle [6, 7]. We define the function $wat : \mathcal{AT} \to \mathbb{R}$ to retrieve the weight of an accessor type, e.g., $wat(getAccessorType(a, i))$ where $getAccessorType(a, i) \in \{an, gn, rn\}$ according to the definition in Table 1.3.

Sensitivity levels of data items. The actor's space, relationships and items, embody the actor's data in OSNs. Using *sensitivity levels*, the associated controllers of an item indicate how much a disclosure of the item would harm them. In what follows we define a set of sensitivity levels that associated controllers can add in their access control policies (see Definition 3). Let

Table 1.3: Accessor types weights

Accessor Type	Numerical value
an	1
gn	0.75
rn	0.50

Table 1.4: Sensitivity levels

Linguistic term	Numerical value
none	0
low	0.25
medium	0.50
high	1

$\mathcal{SL} = \{none, low, medium, high\}$ be the set of *sensitivity level* linguistic terms. The *sensitivity levels* are shown in Table 1.4: the linguistic terms, which are the inputs that are assigned by the associated controllers, correspond to numerical values. We use the function $wsl : \mathcal{SL} \rightarrow \mathbb{R}$ to acquire the numerical value associated to the sensitivity level linguistic term.

Access Control Policies. The associated controllers can define their privacy preferences, where the policy of each controller affects the collaborative decision of viewing and sharing an item. We define an access control policy as follows.

Definition 3. *An* access control policy *is a tuple* $\langle i, c, ct, sl, PER, DEN \rangle$ *where: i)* $i \in \mathcal{I}$ *is the item to which this policy applies; ii)* $c \in$ *is the associated controller who defines the policy over the considered item; iii)* $ct \in \mathcal{CT}$ *is the type of the associated controller—automatically extracted from the corresponding relation in the OSN model; iv)* $sl \in \mathcal{SL}$ *is the sensitivity level of the considered item; v)* PER *and* $DEN \in \mathcal{U}_{\mathcal{AS}}$ *are two accessor specification sets indicating the actors permitted and denied to view the item, respectively.*

We denote the universe of access control policies as $\mathcal{U}_{ACP} \subseteq 2^{\mathcal{I} \times \mathcal{A} \times \mathcal{CT} \times \mathcal{SL} \times \mathcal{U}_{\mathcal{AS}} \times \mathcal{U}_{\mathcal{AS}}}$. For every item $i \in \mathcal{I}$, we use $ACP_i \in \mathcal{U}_{AC\mathcal{P}}$ to denote the set of access control policies of the associated controllers . We denote the access control policy of each associated controller as acp_c, where $c \in \mathcal{A}$. Given ACP_i we use $ACP_i.acp_c.e$ to refer to an element e of the access control policy tuple. Let us illustrate an access control policy with an example as follows: "Alice performs a post (p) (she is the owner) and grants all actors who have a family relationship with her to view the p and denies all her friends to view her post p, with high sensitivity level". Such access control policy for that post p is expressed as: $acp = \langle p, Alice, owner, high, \{\langle family, rn\rangle\}, \{\langle friends, rn\rangle\}\rangle$.

Note that we provide support to explicitly specify permitted and denied actors. This feature is present in OSNs such as Facebook, where actors can, for instance, share an item with their friends, and additionally, explicitly exclude other actors. For instance, consider a policy defined by Bob with: $PER = \{\langle friends, rn\rangle\}$ and $DEN = \{\langle Alice, an\rangle\}$—assuming that the item is shared with Bob's friends and Alice is the actor to be excluded. Note that Alice may, or may not, be friend with Bob. The potential set of actors that may be granted viewing permission is the union of all the permitted actors of all policies for the item, formally, $\bigcup_{acp \in ACP_i} acp.PER$. Every time that an actor in DEN is included in the previous union, a conflict may arise, as it will be the case in the example if Alice is in the permitted set of other associated controller. (We describe the conflict resolution algorithm in Section 2.3).

Note that specifying only permitted actors and marking the other actors in the OSN as denied, or vice versa, is a strictly less expressive choice—in particular the policy above would not be possible to express. So, expressing such cases in our framework is possible, but it may be tedious—the denied and permitted sets may contain a large number of actors. In order to model this in a compact manner, we use a special element $\bot$ that can be included in DEN and represents "all actors not in PER". Likewise, we use the element $\top$ in PER to denote "all actors not in DEN". For instance, a policy with $PER = \{\langle Alice, an\rangle\}$ and $DEN = \{\bot\}$ means "Alice is permitted and anybody else is denied", and with $PER = \{\top\}$ and $DEN = \{\langle Alice, an\rangle\}$ means "Alice is denied and anybody else is permitted". Note that PER and DEN can be empty. Intuitively, polices where PER is not empty and DEN is empty specify only permitted actors, e.g., $PER = \{Alice\}$ and $DEN = \emptyset$ means Alice is permitted and nobody is denied. Note the different with $DEN = \{\bot\}$ where everybody except for Alice is denied. Not specifying denied actors does not mean that everyone can access the item. In particular, in the example above, Alice is the only actor who may be permitted to access the item. Defining $DEN = \emptyset$ simply imposes no restrictions in the set of permitted actors that other associated controllers may allow (in their respective PER sets). For example, consider a policy with $PER = \{Alice\}$ and $DEN = \emptyset$, and a different policy for the same item with $PER = \{Bob\}$ and $DEN = \emptyset$, then the audience of the item is *only* $\{Alice, Bob\}$ (see Section 2.3.1). The intuition behind policies with $PER = \emptyset$ and $DEN \neq \emptyset$ is the inverse of the previous explanation, i.e., they specify only denied actors and leave unspecified the permitted set of actors.

Normalization of Accessor Specifications. It is possible that, within a single access control policy, an actor is permitted and denied access at the same time, thus creating a conflict in the policy. These conflicts may arise explicitly or implicitly. An *explicit conflict* occurs when a concrete actor, group or relationship type is explicitly included in the permitted and denied accessors sets at the same time. *Implicit conflicts* may occur, for instance, if an actor appears explicitly in the permitted accessors set but the denied accessors set includes a group or relationship where the actor is a member of—e.g., if Alice is a friend of Bob and we have $PER = \{\langle \mathit{friends}, rn \rangle\}$ and $DEN = \{\langle Alice, an \rangle\}$. Also, when an actor is a member of two different groups that appear in the denied and permitted accessor sets, respectively—i.e., imagine that Alice belongs to the groups engineers and mathematicians, and we have $PER = \{\langle \mathit{mathematicians}, gn \rangle\}$ and $DEN = \{\langle \mathit{engineers}, gn \rangle\}$. Similarly, if an actor is a member of two different relationship types that appear as permitted and denied, a conflict occurs. Finally, if an actor belongs to a relationship type and a group that appear in different sets, it will cause a conflict, e.g., $PER = \{\langle \mathit{mathematicians}, gn \rangle\}$ and $DEN = \{\langle \mathit{friends}, rn \rangle\}$ would cause a conflict since Alice belongs to both.

To resolve explicit conflicts we check that the sets of permitted and denied accessors are be mutually exclusive, i.e., $PER \cap DEN = \emptyset$.

Resolving implicit conflicts requires looking into the following cases: i) Actors permitted and denied at different hierarchical levels, and; ii) Actors permitted and denied in different groups or relationship types.

Before handling the specific kinds of implicit conflicts, we apply a preprocessing step where we replace pairs $\langle \mathcal{G}, gn \rangle$ with $\langle m_1, gn \rangle, \langle m_2, gn \rangle, \ldots, \forall m_i \in \mathcal{G}$ in the accessor specification sets PER and DEN. Likewise, we replace relationship types pairs such as $\langle \mathcal{R}, rn \rangle$ with their members. As a result we obtain the multisets M_{PER} and M_{DEN}. We use multisets because it is necessary for our conflict resolution strategies to count how many times a pair appears. Note that the pairs in these sets have the type $\mathcal{A} \times \mathcal{AT}$ since we replaced every group and relationship type with their members. Therefore, we can now syntactically identify conflicts by checking the first element of the pairs.

In order to resolve conflicts between actors at different hierarchical levels we apply the *most-specific-takes-precedence* principle [6, 7]. It states that the accessor specification that is more specific should remain. For example, if Bob is in the permitted set as an actor an, and in the denied set because he belongs to a group gn, the strategy removes Bob from the denied set.

Formally, we apply the following principle: "If $\langle a, at_x \rangle \in X$ and $\langle a, at_y \rangle \in Y$ and at_x *is more specific than* at_y, then $X := X \setminus \{\langle a, at_x \rangle\}$ where $X, Y \in \{M_{PER}, M_{DEN}\}$" where the operation \ over multisets discards all occurrences of the elements to remove.

Once we apply the previous step, there might still exist conflicts among groups or relationship types at the same hierarchical level—e.g., if Alice belongs to the groups co-workers and family, and she has permitted one but not the other. To resolve this type of conflict, we apply the *many-takes-precedence* principle [7], i.e., the higher number of positive/negative policies prevail. Formally, given the multisets $X, Y \in \{M_{PER}, M_{DEN}\}$ this principle updates them as follows: $X = X \setminus \{\langle a, at \rangle\}$ if $count(\langle u, at \rangle, Y) > count(\langle a, at \rangle, X)$ where $at \in \{gn, rn\}$, and $count(e, S)$ returns the number of appearances of element e in a multiset S. Note that in the previous strategy we require that the number of elements in one set must be strictly greater than in the other.

Finally, there can still be conflicts if there is the same number of appearances in both multisets. To solve these conflicts we use the *denial-takes-precedence* principle [7]. It simply keeps the pair appearing in the denied accessors set and removes it from the permitted accessors set. Formally, given $at \in \{gn, rn\}$, $M_{PER} = M_{PER} \setminus \{\langle a, at \rangle\}$ if $\langle a, at \rangle \in (M_{PER} \cap M_{DEN})$.

After applying the previous steps in the described order, we add all the elements from M_{PER} to PER and M_{DEN} to DEN to remove any remaining duplicate pairs. The resulting PER and DEN sets are not in conflict. Hence, for the rest of the paper, we assume that the sets PER and DEN are conflict-free.

2.3 Collaborative Access Control

In this section, we introduce our collaborative access control algorithms. These algorithms correspond to two actions that users may perform in the OSN which can potentially involve the controllers types in our model: viewing and sharing. In a nutshell, viewing corresponds to the event of accessing (*view*) to an existing data item. Sharing, on the other hand, consists in selecting an existing item and share a copy in another profile.

2.3.1 Collaborative Access Control: Viewing

Here we present Algorithm 1, an algorithm that produces the list of actors that can view an item. It takes as input three parameters: 1) a set of access control policies ACP_i for the item i; 2) the set of associated controllers for this item C_i, and; 3) the trust graph (TG). As output, it returns a set of *viewers*, i.e., actors that can view item i. First, we include the set of associated controllers C_i in the set of viewers. Thus, modeling that all associated controllers will always be able to view the item. Second, we use the function *normalize*(ACP_i) to resolve internal conflicts within each individual policy as described in the previous section. Also we use an external procedure named *generateAcccesors*(ACP_i) to create a set of all possible viewers from the access control policies (ACP_i) of the item. Concretely, *generateAcccesors*(ACP_i) computes the union of all the actors that appear in the sets of accessor specifications PER and DEN of the policies in ACP_i. Formally, *generateAcccesors*(ACP_i) =

$$\{a \mid (a, at) \in acp.PER, acp \in ACP_i\} \cup$$
$$\{a \mid (a, at) \in acp.DEN, acp \in ACP_i\}$$

The set is created by adding the actors specified in the sets PER and DEN of each policy to ACP_i. The algorithm aggregates the weights of the controller types, accessor types, trust level and sensitivity level indicated in the access control policies defined by the associated controllers according to Equations (1.1) and (1.2) below. Concretely, we use Equation (1.1) for accessor specifications in the PER set, and Equation (1.2) for accessor specifications in the DEN set.

$$\begin{aligned}\texttt{decision\_permit} = \phi_{ct} \cdot wct(acp.ct) + \phi_{at} \cdot wat(getAccessorType(a, acp.i)) +\\ \phi_{tr} \cdot TG.infer(acp.c, a) + \phi_{sl} \cdot wsl(acp.sl)\end{aligned} \tag{1.1}$$

$$\begin{aligned}\texttt{decision\_deny} = \phi_{ct} \cdot wct(acp.ct) + \phi_{at} \cdot wat(getAccessorType(a, acp.i)) +\\ \phi_{tr} \cdot (1 - TG.infer(acp.c, a)) + \phi_{sl} \cdot wsl(acp.sl)\end{aligned} \tag{1.2}$$

The equations depend, among other things, on four *factors*: $\phi_{ct}, \phi_{at}, \phi_{tr}, \phi_{sl}$ (where each $\phi_i \in [0, 1]$) representing the importance we give to each one of the different components of the equation. In particular, ϕ_{ct} affects the weight of controller types, ϕ_{at} affects the weight of accessor types, ϕ_{tr} affects the trust and ϕ_{sl} affects the sensitivity level of the item. In the rest of the paper we omit the factors in our examples for sake of simplifying the presentation.

Note that the `decision_permit` and `decision_deny` equations are present in the algorithm as part of the computation of the variable `decision`. Their value must be computed n times where n ranges from 1 (the owner must always exist) to len (the total amount of associated controllers involved in the decision). The final result is given as $\texttt{decision} = \sum_{n=1}^{len} \texttt{decision\_permit}_n - \texttt{decision\_deny}_n$. If $\texttt{decision} > 0$ then the accessor can access the item, otherwise she cannot. Note that by using our proposed collaborative access control framework, each associated controller of an item has the ability to affect the final decision.

```
input  : $ACP_i$, $C_i$ and $TG$
output: viewers
viewers $\leftarrow C_i$
$normalize(ACP_i)$
set_accessors $\leftarrow generateAcccesors(ACP_i)$
foreach $a \in$ set_accessors do
    decision $\leftarrow 0$
    foreach $acp \in ACP_i$ do
        if $a \in acp.PER$ then
            decision $\leftarrow$ decision + $\phi_{ct} \cdot wct(acp.ct)+$
              $\phi_{at} \cdot wat(getAccessorType(a, acp.i))+$
              $\phi_{tr} \cdot TG.infer(acp.c, a)+$ $\phi_{sl} \cdot wsl(acp.sl)$
        end
        else if $a \in acp.DEN$ then
            decision $\leftarrow$ decision $-$ $\phi_{ct} \cdot wct(acp.ct)+$
              $\phi_{at} \cdot wat(getAccessorType(a, acp.i))+$
              $\phi_{tr} \cdot (1 - TG.infer(acp.c, a))+$ $\phi_{sl} \cdot wsl(acp.sl)$
        end
    end
    if decision $> 0$ then
        viewers $\leftarrow$ viewers $\cup \{a\}$
    end
end
```

Algorithm 1: Viewing

Example 1. *Consider that Alice performs a post p and mentions her family members Bob and Carol. Alice is the owner whereas Bob and Carol are the stakeholders. Their access control policies are,* $ACP_p =$

$$\{\langle p, Alice, owner, low, \{\langle family, rn\rangle\}, \{\langle friends, rn\rangle\}\rangle,$$
$$\langle p, Bob, stakeholder, medium, \{\langle co\text{-}worker, rn\rangle\}, \emptyset\},$$
$$\langle p, Carol, stakeholder, low, \{\langle friends, rn\rangle\}, \emptyset\}.$$

Consider now an actor (David), who is a friend of Alice and Carol. In ACP_p *Alice denies her friends whereas Carol allows her friends. In this scenario, the positive and negative authorizations about David's access create a conflict. The permitted decision value (*`decision_permit`*) is aggregated from Carol's acp which affects David's access as he is a friend of her. Carol has a low sensitivity level for the post p. On the other hand, the algorithm computes the value of a denied decision from acp of Alice which affects David's access as he is her friend. Alice also has a low sensitivity level for the post p. owner and stakeholder are* *weighted 1 as defined in Table 1.2. For the purpose of this example, let Alice define a trust value of 0.75 for David whereas Carol sets a trust value of 0.5 for David. According to these trust values and the associated controllers' privacy policies, the value of* `decision_deny` = 2 *based on Equation* (1.2) *whereas the value of* `decision_permit` = 2.25 *based on Equation* (1.1). *So, David will have access to view the post.*

2.3.2 Collaborative Access Control: Sharing

Our second algorithm produces a set of actors that can share an item which has been previously posted (see Algorithm 2). We call *disseminators* the actors that have the right to share an item. The event of sharing an item consists in copying an already posted item and placing it in the disseminators space. The shared item placed in the disseminator's space is different from the original item in the owner space. Note that a shared item has one owner, one originator, and zero or more stakeholders—because we restrict sharing to the disseminator space.

This algorithm, as opposed to Algorithm 1, includes two phases: 1) filtering viewers in potential allowed disseminators and potential denied disseminators, based on the trust that the associated controllers have for each viewer (specified sharing policies, Definition 4 below), and; 2) an aggregation-based method similar to that of Algorithm 1 to decide whether the conflicting actor—i.e., an actor permitted by some associated controllers and denied by other associated controllers—might become a disseminator.

Sharing Policies. For sharing, each associated controller specifies a trust threshold that determines how much the minimum value of trust between her and the viewer has to be in order to allow the sharing action.

Definition 4 (Sharing Policies). *We define a* sharing policy *as a tuple* $\langle i, c, trc\rangle$ *where: 1)* $i \in \mathcal{I}$ *is the item to which the policy applies; 2)* $c \in \mathcal{A}$ *is the associated controller who defines the sharing policy, and; 3)* $trc \in \mathbb{R}$ *is a numerical value specifying the trust threshold for which sharing the item* i *is permitted by* c.

For every item $i \in \mathcal{I}$, $SP_i \in \mathcal{U}_{\mathcal{SP}}$ is the set of sharing policies (with $\mathcal{U}_{\mathcal{SP}} \subseteq 2^{\mathcal{I}\times\mathcal{A}\times\mathbb{R}}$). The numerical value in *trc* is obtained from a $\mathcal{TLT}$ that the associated controller selects when defining the policy—Table 1.1 contains the equivalences between TLTs and their corresponding numerical value. Given a sharing policy *sp* we use *sp.e* to refer to an element *e* of the sharing policy tuple, e.g., *sp.trc* refers to the trust threshold of the sharing policy *sp*.

Algorithm 2 takes as input four parameters: 1) a set of access control policies ACP_i for the given item *i*; 2) the trust graph (*TG*); 3) a set of `viewers` (computed using Algorithm 1), and; 4) the sharing thresholds associated to the item i, SP_i. As output, it returns the set of actors that can share the item. In what follows we explain the two phases of the algorithm in detail.

Phase 1). As mentioned earlier, SP_i contains the sharing policies for item i. These policies determine the minimum value of trust between the associated controller and the actor who might share the item. We use an external procedure named *filterActors* (i,TG,SP_i,`viewers`) to split the potential disseminators into actors who do (`permit_sharing`) and do not (`deny_sharing`) fulfill the sharing policies SP_i set in advance by each associated controller. The pseudo-code of the procedure is shown below:

```
foreach $a \in$ viewers do
    foreach $sp \in SP_i$ do
        $tr = TG.infer(sp.c, a)$;
        if $tr \geq sp.trc$ then
            permit_sharing $\leftarrow$ permit_sharing $\cup \{c\}$;
        else
            deny_sharing $\leftarrow$ deny_sharing $\cup \{c\}$;
        end
    end
end
```

Procedure filterActors(i,TG,SP_i,`viewers`)

Given an item i, the above procedure includes a viewer v in the set of potentially permitted disseminators (`permit_sharing`), if an associated controller *c* has specified a sharing policy *sp* that includes a trust level lower than the trust the associated controller defined for the viewer, i.e., $TG.infer(sp.c, a) \geq sp.trc$. Otherwise the actor is included in the set of potentially denied disseminators (`deny_sharing`). Note that, since `permit_sharing` and `deny_sharing` are sets, each viewer can appear at most once in each set.

Phase 2). When a conflict arises among the associated controllers to allow or refuse the sharing action of the item i, for all involved associated controllers we have to differentiate between two cases: 1) when the trust of the actor who might share the item is equal or higher than the sharing threshold (Equation (1.3)), and; 2) when the trust of the actor who might share the item is lower than the sharing threshold (Equation (1.4)).

$$\texttt{decision\_permit} = \phi_{ct} \cdot wct(acp.ct) + \phi_{sl} \cdot wsl(acp.sl) \tag{1.3}$$

$$\texttt{decision\_deny} = \phi_{ct} \cdot wct(acp.ct) + \phi_{sl} \cdot wsl(acp.sl). \tag{1.4}$$

Contrarily to the Viewing algorithm, both Equations (1.3) and (1.4) have only two components, i.e., the weight of the controllers types and the sensitivity level of the associated controllers with respect to the item to be shared. This is because the set of actors that can share an item already had privileges to access it (viewers). This difference directly impacts the structure of the decision formula in such a way that the trust is used as a filter to know in advance if an actor is willing to share an item (there is no reason to include the accessor's weight as there are no accessors involved in the algorithm). As for the first two equations, we also introduce factors, two in this case: ϕ_{ct} and ϕ_{sl} (where each $\phi_i \in [0, 1]$) representing the importance we give to each one of the components of the formula. Concretely, ϕ_{ct} affects the weight of the controller types whereas ϕ_{sl} affects the sensitivity level of the item. Such factors may be used in a fine grained manner to prioritize one component over the other, in the same way as for the first two equations.

As mentioned earlier, the set of viewers always includes the associated controllers C_i. So, they are considered by the Sharing algorithm. As opposed to the Viewing algorithm, where the associated controllers are always part of the permitted actors to view the item, the Sharing algorithm treats the associated controllers as any other viewer. Therefore, it is not guaranteed that an associated controller can share an item unless the sharing policies specified by the rest of the associated controllers permit it.

Example 2. *The result of running Algorithm 1 in Example 1 was that David is in the* `viewers` *set. We run the Sharing algorithm in order to determine whether David can be a disseminator or not. The sharing policies of the associated controllers are,* $SP_p = \{\langle p, Alice, 1\rangle, \langle p, Bob, 0.50\rangle, \langle p, Carol, 0.25\rangle\}$*. Remember that in Example 1, Carol defined a trust value of 0.5 for David, and Alice defined a trust value of 0.75 for David. We assume 0.25 is the returned value of the infer function due to the indirect connection between Bob and David. In this scenario, David fulfills Carol's sharing policy but he does not satisfy the minimum value of trust set by Alice and Bob generating then a conflict and executing the Sharing algorithm. According to the associated controllers' privacy policies, the value of* `decision_permit` *is equal to 1.25 based on Equation* (1.3)*. On the other hand, the value of* `decision_deny` *is equal to 2.75 based on Equation* (1.4)*, which means the final result is to deny David to share the post.*

2.4 Computational Complexity

The algorithms presented here have linear time complexity. The Viewing algorithm with respect to the size of the set of accessors (potential viewers), and the Sharing algorithm with respect to the size of the input set of viewers.

```
input  : $ACP_i$, $TG$, viewers and $SP_i$
output: disseminators
permit_sharing, deny_sharing ← $filterActors(TG, SP_i$ , viewers)
foreach $a \in$ viewers do
    decision ← 0
    foreach $acp \in ACP_i$ do
        if $a \in$ permit_sharing then
            decision ← decision
              $+ \phi_{ct} \cdot wct(acp.ct) + \phi_{sl} \cdot wsl(acp.sl)$
        end
        else if $a \in$ deny_sharing then
            decision ← decision
              $- \phi_{ct} \cdot wct(acp.ct) + \phi_{sl} \cdot wsl(acp.sl)$
        end
    end
    if decision > 0 then
        disseminators ← disseminators ∪ $\{a\}$
    end
end
```

Algorithm 2: Sharing

In practice, these boundaries are never large enough to noticeably affect the performance. We explain the complexity of both algorithms in more detail below.

Algorithm 1 Let n be an upper bound in the number of input access control policies and an upper bound m in the size of the set of accesors—i.e., the sum of the sizes of the sets *PER* and *DEN*. The linear complexity of Algorithm 1 is $O(n \times m)$. This result trivially follows from the fact that, for each policy, it is necessary to go through all the actors included in *PER* and *DEN*. Very often the set of policies n is not very large [17] (it can thus be regarded as constant). Therefore, the time complexity of the algorithm is linear with respect to m.

For a practical implementation, the main drawback of this result is that the upper bound on the size of the sets *PER* and *DEN* might be large. In particular, actors with many friends, or who belong to vastly populated groups, may include in their policy sets *PER* and *DEN* many actors. For instance, some studies show that, on average, Facebook users have around 300 friends [32]. Therefore, optimizations that avoid checking all the actors in *PER* and *DEN* can have a great impact in the performance of the algorithm.

Algorithm 2 Given an upper bound j on the size of the viewers set, and an upper bound k on the number of input sharing policies, the time complexity of Algorithm 2 is $O(j \times k)$. Note that the number of sharing policies will usually be small [17], therefore the algorithm has linear time complexity with respect to k.

Though the set of viewers may be large, this algorithm allows for a simple optimization which reduces its time complexity to constant. In many OSNs, it is unnecessary to compute *a priori* the set of users that can share a post. It is possible to execute the algorithm for a single actor on demand. For instance, when the actor is about to view the item, the algorithm may be run for this particular actor. As a consequence the factor j in the previous complexity is reduced to a constant size of 1. Note that the algorithm is only executed the first time an actor views a post.

3 Evaluation

To evaluate how our proposed algorithms behave, we implemented our solution in Python and executed it on a MacBook Air with 2.2 GHz Intel Core i7 CPU and 8 Gb of RAM. We computed all possible combinations of trust, sensitivity level and controller types for the Viewing algorithm as well as the sensitivity level for Sharing algorithm of all the associated controllers: owner, contributor, originator and stakeholders. It is important to recall that there can be at most one contributor, one originator and zero or more stakeholders where the owner is the only mandatory one. In the particular case of the Viewing algorithm, each one of these associated controllers might have 5 possible trust values, 4 sensitivity levels and can define 3 different accessors types for the accessors whereas for the Sharing algorithm, the associated controllers might only have 4 sensitivity levels defined. Additionally, each one of the associated controllers may permit or deny the access or the sharing action for an item.

We split the evaluation into two main parts corresponding to Viewing and Sharing. We omit the factors in our evaluation as the objective is to see the interplay of our components and not how the factors affects the outcome.[5] Additionally, and for the sake of simplicity, in what follows we describe the evaluation of the cases where the distance between associated controllers is 1 (see Table 1.2). Though the remaining cases are not explicitly presented, they can be found in our implementation. All the source code of our implementation are publicly accessible online [1].

[5] The factors may be considered as parameters that fine tune our decision algorithms, providing a range of decisions from more conservative to more liberal.

Finally, as a consequence of the collaborative nature of our proposal, both Viewing and Sharing algorithms generate a so-called *flipping point*. This point represents the number of associated controllers who are needed in order to revoke an action. For example, let assume that Alice (the owner of an item) does not want Bob to access to the item, but other associated controllers want to let him access to it. The question is then how many associated controllers are needed in order to revoke Alice's policy and let Bob access to the item. We calculated that point for both algorithms as explained later in this section assuming all involved factors to be 1.

Viewing. In all the figures related to Viewing, i.e., Figures 1.2 to 1.5, the root node denotes the associated controllers for whom the experiment is running. The first level means the possible trust values. The second level symbolizes the possible values for the sensitivity levels. The third level stands for the accessor types possibilities whereas finally, the leaves represent the output of the equation being executed according to the experiment.

Figure 1.2 shows all the outputs when there is only one associated controller, either an owner or a stakeholder—both associated controllers generate the same values. For this experiment we created an access control policy containing an arbitrary accessor when she is in the permitted set (PER) of the owner/stakeholder.

Similarly, we run the same experiment for the contributor/originator as both generate the same values when the distance is 1, and the outputs can be seen in Figure 1.3. As expected, the values are the same as in Figure 1.2 with a difference of -0.5—which is the only difference between owners/stakeholder and contributor/originator (see Table 1.2).

It is straightforward to compute the values for the same associated controllers when the accessor is in the denied set. We generated all possible values for Equation (1.2) and the results are shown in Figures 1.4 and 1.5 for the owner/stakeholder and contributor/originator, respectively. From the results, it can be stated that when the owner/stakeholder defines the accessor in the DEN set and the values of the trust, sensitivity level and the access type are none, high and AN respectively, the decision value achieves its maximum number (4.0) as expected and the similar behavior is seen for the contributor/originator achieving 3.5 as its maximum value under the same settings. Finally, we included in Figures 1.7 and 1.8 corresponding to the combinations where the contributor distance ≥2 and the viewer is in the PER and DEN respectively.

As an example, let us suppose that in our collaborative OSN, there are two associated controllers: an owner and an originator. The setting of the values

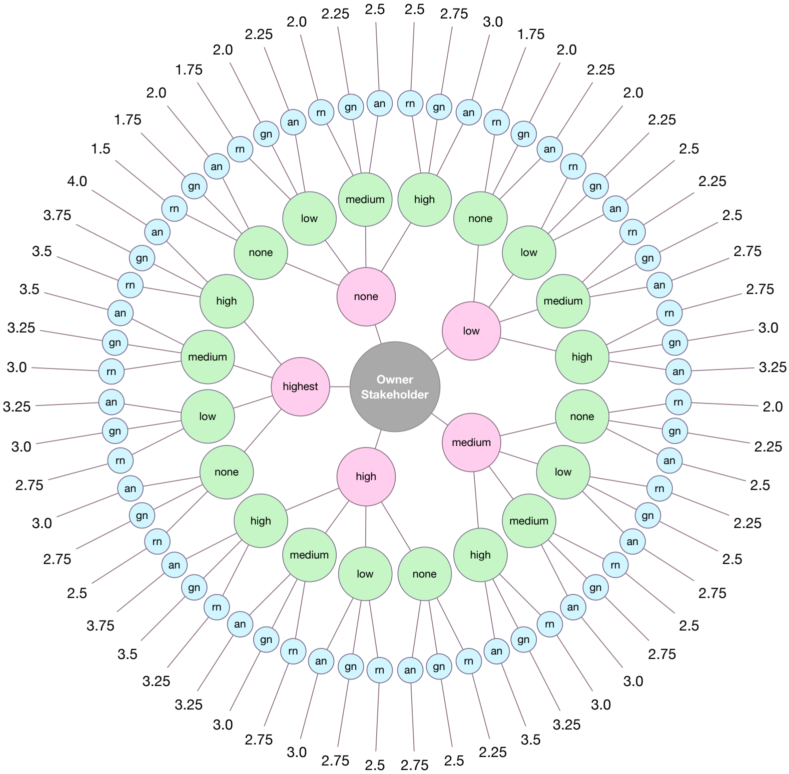


Figure 1.2: Viewing: Decision values for the owner/stakeholder with $(\phi_{ct}, \phi_{at}, \phi_{tr}, \phi_{sl}) = 1$ and the accessor $\in PER$

regarding the owner for a particular item and a given accessor who is in her permitted set are: trust=*highest*; sensitivity_level=*low*; accessor_type=*an*. On the other hand, the originator has the same accessor in the denied set in such a way that there is a conflict and her settings are: trust=*none*; sensitivity_level=*medium*; access_type=*gn*. Finally, the output of Equation (1.1) is 2.75 whereas the output of Equation (1.2) is 3.25. The final decision is that the accessor can access to the item (since $3.25 - 2.75 > 0$).

Sharing. In Figures 1.6a and 1.6b, the root node denotes the associated controllers for whom the experiment is running. The first level denotes the possible values for the sensitivity levels and the leaves represent the output of Equations (1.3) and (1.4) respectively.

To evaluate how Equations (1.3) and (1.4) behave, we calculated all the possible values that these equations can generate when the viewer is in either

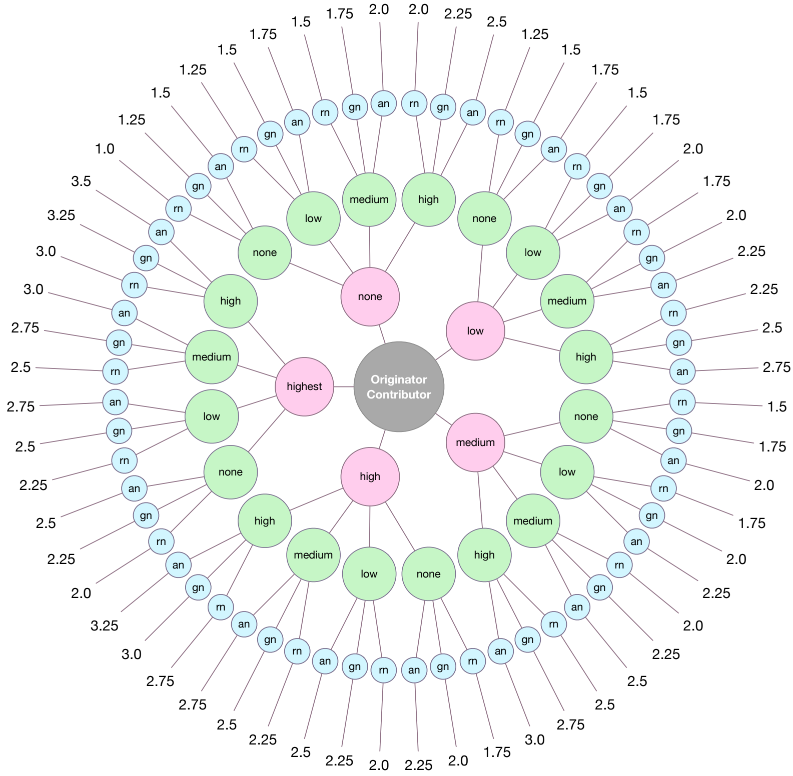


Figure 1.3: Viewing: Decision values for the contributor/originator with (ϕ_{ct}, ϕ_{at}, ϕ_{tr}, ϕ_{sl}) = 1; accessor ∈ *PER*; and distance=1

permit_sharing or deny_sharing sets generated by the external procedure *filterActors*.

In particular, Figures 1.6a and 1.6b depict the decision values when the viewer is in the permit_sharing for the owner/stakeholder (Figure 1.6a) and for the contributor when the distance is equal to 1 (see Figure 1.6b). As expected, the value of the leaves only differ on 0.5 (see Table 1.2). It is worth mentioning that we have not included Figures when the viewer is in the deny_sharing because in the Sharing algorithm they produce the same ones. In 5, we run the same experiments and generated the same figures for the rest of the cases, i.e., when the distance of the originator is greater than 2, and all the cases for the originator, i.e., when the *infer* function returns either 0.25 or 0.5.

As an example, let us suppose that in our collaborative OSN, there are two associated controllers: an owner and a contributor. The sensitivity level

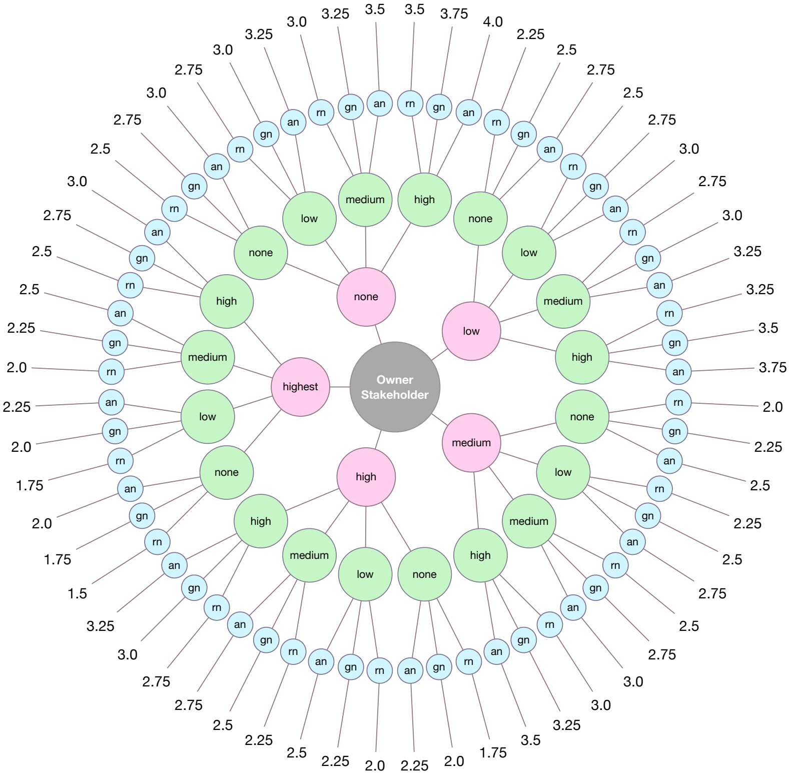


Figure 1.4: Viewing: Decision values for the owner/stakeholder with $(\phi_{ct}, \phi_{at}, \phi_{tr}, \phi_{sl}) = 1$ and the accessor $\in DEN$

regarding the owner for a particular item and a given viewer who is in her permitted set is low. On the other hand, the contributor defined her sensitivity level as medium and she has the same viewer in the denied set thus, generating a conflict. In both cases, the viewer fulfills the sharing threshold defined by the associated controllers. With these values, the output of Equation (1.3) is 1.25 whereas the output of Equation (1.4) is 1. The decision is that the viewer can finally share the item, since $1.25 - 1 > 0$.

Flipping Point. We also computed the *flipping point* to determine how many associated controllers are needed in order to revoke the decision taken by another (set of) associated controller(s). Note that this is a combinatorial problem since there can exist a large number of stakeholders—the upper bound is given by the number of actors in the OSN apart from the owner, who can grant or deny privileges for the viewing or sharing actions to the

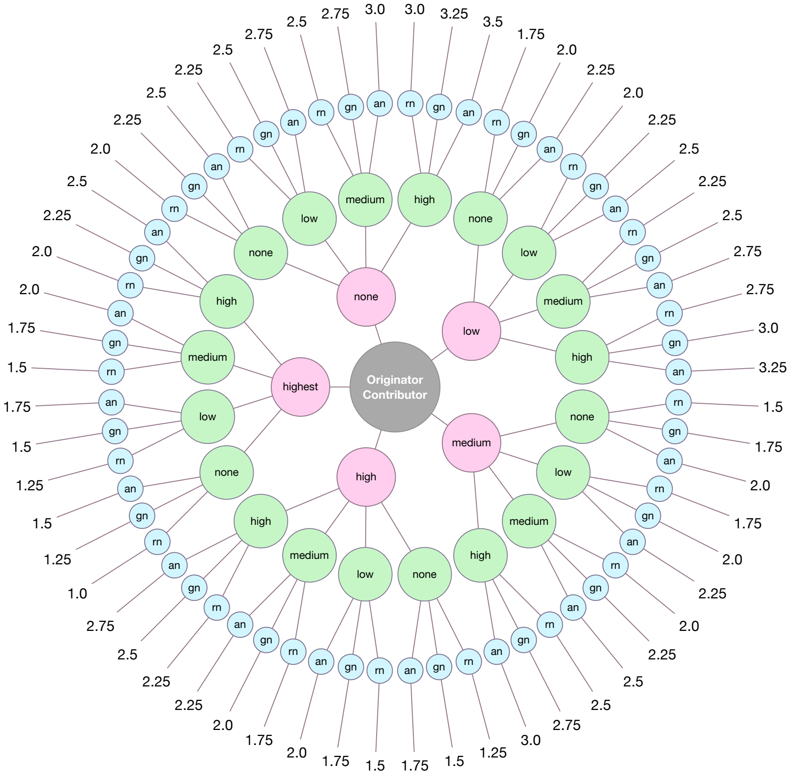
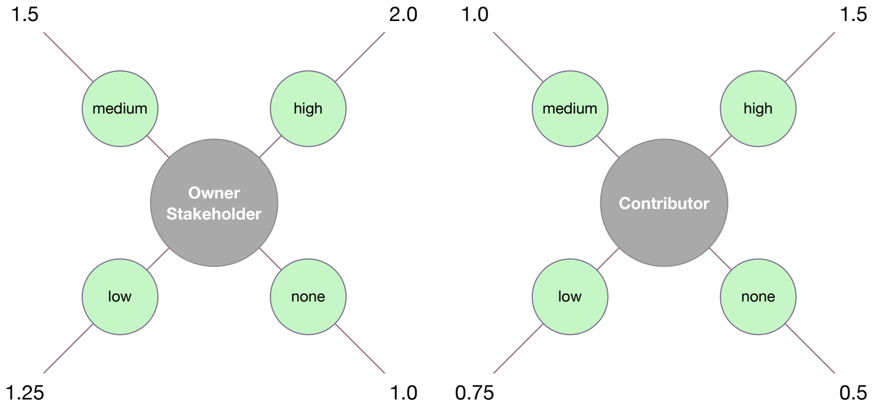


Figure 1.5: Viewing: Decision values for the contributor/originator with (ϕ_{ct}, ϕ_{at}, ϕ_{tr}, ϕ_{sl}) = 1; accessor ∈ *DEN*; and distance=1

actors. Hence, based on the experiments we carried out previously in this section, we are providing a general way to compute such a flipping point.

We first focus on the Viewing algorithm with one of the simplest scenarios, i.e., the owner and a stakeholder who wants to revoke the owner's decision. We computed the frequency of all the outputs (second row of Table 1.5) in such a way that the maximum value for an owner with respect to a particular item is 4 and the frequency is 1 (as it only appears once in Figure 1.2). The third row shows how many different ways a stakeholder might revoke the initial decision, i.e., how many possible ways the stakeholder can get a number greater than the output given in the second row. Finally, in the last row, we computed the probability that an owner's decision might be revoked by one stakeholder. For example, when the owner achieves a 3.5 as output of the formula (e.g., trust=*highest*; sensitivity_level=*high*; access_type=*rn*), there is a 5% of probability that the stakeholder revokes her decision.



(a) owner/stakeholder **(b)** contributor with distance=1

Figure 1.6: Sharing: Decision values with $(\phi_{ct}, \phi_{sl}) = 1$ and viewer ∈ `permit_sharing`

Table 1.5: Viewing baseline probability of revoking the owner or a stakeholder's initial decision for only one stakeholder or the owner respectively

Viewing — Stakeholder											
Output	4	3.75	3.5	3.25	3.0	2.75	2.5	2.25	2.0	1.75	1.5
Frequency	1	2	4	6	9	10	10	8	6	3	1
Revocation number	0	1	3	7	13	22	32	42	50	56	59
Baseline Probability	0%	1.6%	5%	11.6%	21.6%	36.6%	53.3%	70%	83.3%	93.3%	98.3%

Once we have that baseline probability, it is pretty straightforward to compute such probability for any number of associated controllers. There are two cases, when the accessor is either in the *PER* or in the *DEN* sets of the associated controllers. The order matters when computing the baseline probability, so we should first generate the associated controllers who have the intended accessor in their *PER* set and then in their *DEN* set. The reason for this is that the *PER* set directly affects the output (row 1) and the frequency (row 2), whereas the *DEN* set only affects the revocation number (row 3).

On the one hand, if the accessor is in the *PER* set, the range of the outputs is computed by multiplying the max (4.0) and min (1.5) values by the number of stakeholders plus the owner. Finally, both the frequency and the revocation number should be recalculated to obtain the probability. As an example, let us suppose that there are 3 stakeholders plus the owner who grant a actor to access to an item. Then the range of the output will go from 6 to 16 by steps of 0.25. In this scenario, it is impossible for only one stake-

Table 1.6: Viewing baseline probability of revoking the owner's initial decision for one contributor/originator when distance is 1

Viewing — Contributor/Originator													
Output	4	3.75	3.5	3.25	3.0	2.75	2.5	2.25	2.0	1.75	1.5	1.25	1.0
Frequency	1	2	4	6	9	10	10	8	6	3	1	0	0
Revocation number	0	0	0	1	3	7	13	22	32	42	50	56	59
Baseline Probability	0%	0%	0%	1.6%	5%	11.6%	21.6%	36.6%	53.3%	70%	83.3%	0%	0%

Table 1.7: Sharing baseline probability of revoking the owner or a stakeholder's initial decision for only one stakeholder or the owner respectively

Sharing — Stakeholder				
Output	2.0	1.5	1.25	1.0
Frequency	1	1	1	1
Revocation number	0	1	2	3
Baseline Probability	0%	25%	50%	75%

holder to revoke the owner's decision (note that the maximum value that a stakeholder can achieve is 4).

On the other hand, if the accessor is in the *DEN* set, both the range of the outputs and the frequency remain the same but the revocation number must be recalculated by adding the number of different options by which the stakeholder may get a number greater than the outputs.

We also computed the same baseline probability when there is an owner and one contributor/originator when the distance is equal to 1. The results can be seen in Table 1.6. The same strategy regarding the *DEN* and *PER* sets explained above is also applied in this table. However, since our framework can only have one contributor and one originator, the possibilities are simplified to the combinations of 3 elements, i.e., if the accessor is in the *PER* or *DEN* sets of the owner, the contributor or the originator. In 5 we carried the same experiment fixing the distance ≥ 2 (see Table 1.11).

From the above results regarding the Viewing algorithm, we conclude that whenever there are 5 associated controllers (remember that the owner is mandatory), if 4 of them have a different opinion than the other one, the decision will be definitely revoked no matter if that actor has a strong opinion (sensitivity) about the item.

We carried out the same analysis for the Sharing algorithm. The results can be seen in Table 1.7 for the stakeholder/owner, Table 1.8 for the contributor when the distance is 1, and Table 1.9 for an owner when her weight is

Table 1.8: Sharing baseline probability of revoking the owner's initial decision for one contributor when distance is 1

Sharing — Contributor						
Output	2	1.5	1.25	1.0	0.75	0.5
Frequency	1	1	1	1	0	0
Revocation number	0	0	1	1	2	3
Baseline Probability	0%	0%	25%	25%	0%	0%

Table 1.9: Sharing baseline probability of revoking the owner's initial decision for one originator when her weight is 0.75

Sharing — Originator					
Output	2	1.5	1.25	1.0	0.75
Frequency	1	1	1	1	0
Revocation number	0	1	1	2	3
Baseline Probability	0%	25%	25%	50%	0%

0.75. For completeness, we included in 5 the baseline probability computation when the contributor's distance is ≥ 2 (Table 1.12) and when the originator's weight is 0.25 (see Table 1.13).

It is interesting to see in this case that if an owner achieves the maximum value (2), then in the best case scenario, a stakeholder plus one more associated controllers are needed to revoke the owner's decision. On the other hand, in the worst case scenario (i.e., all associated controllers achieve the minimum amount in the decision) two stakeholders would be needed to revoke the owner's decision. Finally, while contributors and originators are the less powerful associated controllers in such a way that they can only achieve a 1.5 in the best case scenario, they can be crucial when there only are a few stakeholders involved in the sharing decision.

3.1 Proof-Of-Concept Implementation

Diaspora belongs to the family of decentralized OSNs. In such OSNs there is no single entity where all information is stored. Instead, they consist of independent nodes which host all the information of the social network. Diaspora nodes are called *pods*. A pod is basically a server which host an instance of Diaspora's source code and its own database.

Table 1.10: Comparison between Facebook, Diaspora, CAC Framework and our Diaspora implementation.

Components	Facebook	Diaspora	CAC Framework	Proof-of-Concept
Controllers Types	*owner*	*owner*	*owner, stakeholder, contributor, originator*	*owner, stakeholder*
Acessors Types	*rn, gn, an*	*rn*	*rn, gn, an*	*rn*
Sensitivity Levels	∅	∅	*none, low, medium, high*	*none, low, medium, high*
Trust	∅	∅	*none, low, medium, high, highest*	*none, low, medium, high, highest*

We deployed a particular instance of our approach in Diaspora [1]. Since our main goal is to demonstrate that our approach can be deployed in a real world application, and for the sake of simplicity, we did not use the decentralized architecture that Diaspora provides. Instead, we deployed our own pod on a MacBook Pro with 2.9 GHz Intel Core i5 CPU and 8 Gb of RAM.

Table 1.10 depicts the differences between Facebook, Diaspora, our theoretical model (framework), and our proof-of-concept implementation. We comparison criteria are with respect to what can be expressed in each one of them (and what was implemented). As expected, our framework is more general than Facebook and Diaspora since it considers all the controller and accessors types, and it allows for a more fine-grained decision concerning permitting/denying access to view/share an item.

Since our framework is more general, Facebook's privacy settings may be modelled in our framework. This is done by giving suitable values to the factors of our algorithms: assign 0 to the factors of accessor type (ϕ_{at}), sensitivity level (ϕ_{sl}) and trust (ϕ_{tr}), and set the weight of all the associated controllers to 0, except for the owner whose weight is 1.

Diaspora does not allow to define sensitivity levels with respect to an item, nor trust between users, we extended Diaspora in order to include such notions. Due to the way Diaspora is implemented, our proof-of-concept implementation only includes the *owner* and *stakeholder* controllers types and accessor type *rn*. Implementing *contributor* and *originator* controller types is not possible because Diaspora does not provide the feature of posting in someone else's profile. Moreover, the accessor types *gn* and *an* are not performed due to other Diaspora constraints: 1) no distinction between relationships and groups, and; 2) social activities such as posting, sharing, etc., are defined over relationships.

Usability. In our implementation we provide a natural and user-friendly way to define privacy settings, having default values that favour the privacy of the users—5 shows the different interfaces of our implementation. Moreover, our privacy setting interface allows fine-grained control for both data and accessors as well as it allows to define both the permitted and denied set of accessors either by using the defined relationships, called "aspects" in Diaspora, such as “family” or “friends”, or even more general relationships such as “Everyone” or “Nobody” (see Figure 1.11). The associated controllers can also determine the sensitivity level of shared items based on their contents as well as the trust level of each relationship and the trust level for users who do not belong to any relationship (see Figure 1.12).

Our pod allows the associated controllers to specify their privacy settings regarding the sharing action (see Figure 1.12). Not all the associated controllers and accessors who have privileges to view the post have the rights to share it. Consequently, in our pod the share button only appears to associated controllers and viewers who have the right to disseminate the post according to a cooperative decision obtained by executing Algorithm 2.

We adapted, by modifying the controller and the accessor types, Examples 1 and 2 to work in Diaspora. As expected, in Example 1, David was able to view the Alice’s post whereas in Example 2, he cannot share it. Besides, we also adapted, by having *rn* as accesor type, all the experiments presented in the paper, and achieved the same outputs as in Figures 1.2, 1.4 and 1.6a, where the owner and a stakeholder are involved.

4 Related Work

In access control models for OSNs, we can distinguish between two main approaches: 1) mechanisms which assume that the information is governed by a single user, e.g., [5, 11, 15], and; 2) mechanisms where a collaborative decision regarding the information is made, e.g., [17, 18, 23, 26, 27, 28, 35]. In the following we only focus on the second approach and we analyze the most relevant proposals published on this topic.

Squicciarini et al. [26, 27] provide a novel collective privacy mechanism for content sharing among users in OSNs. The paper considers that the privacy control of the shared content is co-owned by multiple users, so each stakeholder may separately specify her own privacy settings for the shared data and thus, a voting algorithm to enable collective enforcement for shared data is used. However, in their algorithm only the winners of the voting algorithm control who can access the data, instead of harmonizing all stakeholders’ privacy preferences.

Carminati and Ferrari [3] 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. In contrast, our work proposes a formal collaborative model to manage viewing and sharing of shared items in OSNs, in addition to fine-grained policy specification scheme.

Similarly, Such and Criado [28] propose a mechanism to resolve multiparty conflicts based on the willingness of each associated controller to give access. However, once again, in this work there is no collaborative decision given that if one user has high willingness and another one low willingness, only the former will determine the final access.

Another proposal of a policy-based approach to control access to shared data in OSNs is given in [34]. In this case, the owner of the content is allowed to specify policies for the content she uploads and other users (called trusted co-owners, who previously must be invited by the owner) can edit such a policy. In our proposal we use the same concept in the sense that the owner has to mention users (stakeholders) to be part of the collaborative decision. However, stakeholders in our work do not directly edit the owners policy, and instead we consider their access policies to calculate the decision.

Xu et al. [35] propose a collaborative privacy management mechanism where the collective decision is made by the user who wants to post data (the owner, who is responsible for gathering feedback from other involved users). Though trust values are used to indicate how much influence a user's opinion will have on the aggregated decision, the owner has full decision power on who should access the item.

The approach proposed by Xu et al. [16] is similar to our work in the sense that it offers a systematic solution for detecting and resolving privacy conflicts for collaborative data sharing in OSNs. However, their approach needs a negotiation mechanism to solve the privacy policies conflicts before the access privileges are computed. In order to fix that issue, they improved their work by enhancing a policy specification scheme and a voting-based conflict resolution mechanism [17]. Nevertheless, the conflict resolution strategy presented in such work is selected by the data owner which leads to a unilateral decision, i.e., without considering the privacy preferences of other associated controllers involved. Our work could be seen as an extension as the one presented in [17]: all the associated controllers are taken into consideration for the collaborative decision, so we are indeed giving a truly collaborative access control framework.

Based on the multiparty access control model presented in [17], Vishwamitra et al. [33] introduced a model that allows each involved user in a photo to independently decide whether some personally identifiable information in the photo is shown or blurred. In our scenario, the collaborative decision protects all the items, including photos, from being viewed or shared.

Controlling the content of the item is outside the scope of this article.

Gay et al. [13] provided fine-grained privacy mechanism to control over sharing and re-sharing, and the distribution of re-shared messages in decentralized OSN. Similar to our work, they also base their enforcement of privacy policies on ReBAC. Despite this apparent similarity, our access control policy has more fine-grained features such as the possibility to define an explicit denied set and accessor specification policies. In their work, only trust is used to determine which users, among those who has already received the item, are allowed to propagate this item. In contrast, we apply trust as one of four components in the process of computing the collaborative decision regarding who can view or share a given item. That is, contrary to our approach, the proposal in [13] doses not consider the users associated with the item as co-controllers.

See [22, 24, 25] for a survey on state-of-the-art collaborative access control systems for OSN.

5 Conclusion

We presented a collaborative access control framework for OSNs that collectively achieves a decision about who should (not) access, or (not) share, an item. The decision is based on the privacy settings of all concerned associated controllers, i.e., owner, originator, contributor and stakeholder(s). This is done by taking into account the following four aspects: trust relationship between users, sensitivity level of the users with respect to the concerned item as well as different weights for both the controller types and accessor types. We proposed a Viewing and a Sharing algorithm for taking such a decision about the items. We also evaluated them by generating all combinations of the components, and provided a proof-of-concept implementation in the open source social network Diaspora.

Concerning the correctness of our solution, different decisions could have been taken depending on whether one might want to privilege privacy over utility or vice-versa. This trade-off between privacy and utility may be stretched or relaxed by playing with the factors we have defined, and that are assigned to each one of the components associated with the different privacy setting aspects upon which our decision algorithms are based on. We plan to study and experiment with those factors in future work.

Bibliography

[1] H. Alshareef, R. Pardo, G. Schneider, and P. Picazo. A Collaborative Access Control Framework for Online Social Networks, Feb. 2019.

[2] K. Carley. A theory of group stability. *American sociological review*, pages 331–354, 1991.

[3] B. Carminati and E. Ferrari. Collaborative access control in on-line social networks. In *CollaborateCom*, pages 231–240, 2011.

[4] J. Caverlee, L. Liu, and S. Webb. Socialtrust: tamper-resilient trust establishment in online communities. In *JCDL*, pages 104–114, 2008.

[5] Y. Cheng, J. Park, and R. Sandhu. Relationship-based access control for online social networks: Beyond user-to-user relationships. In *PASSAT*, pages 646–655. IEEE, 2012.

[6] S. De Capitani Di Vimercati, S. Foresti, P. Samarati, and S. Jajodia. Access control policies and languages. *International Journal of Computational Science and Engineering*, 3(2):94–102, 2007.

[7] S. D. C. di Vimercati, P. Samarati, and S. Jajodia. Policies, models, and languages for access control. In *DNIS*, pages 225–237, 2005.

[8] Diaspora. Diaspora. `https://joindiaspora.com`, 2016. [Available Online].

[9] S. L. Feld. The focused organization of social ties. *American journal of sociology*, 86(5):1015–1035, 1981.

[10] P. W. Fong, M. Anwar, and Z. Zhao. A privacy preservation model for facebook-style social network systems. In *ESORICS*, pages 303–320, 2009.

[11] P. W. Fong and I. Siahaan. Relationship-based access control policies and their policy languages. In *SACMAT*, pages 51–60, 2011.

[12] C. Gates. Access control requirements for web 2.0 security and privacy. *IEEE Web*, 2(0), 2007.

[13] R. Gay, J. Hu, H. Mantel, and S. Mazaheri. Relationship-based access control for resharing in decentralized online social networks. In *FPS*, pages 18–34, 2017.

[14] J. A. Golbeck. *Computing and applying trust in web-based social networks.* PhD thesis, 2005.

[15] J. Grossklags, N. Christin, and J. Chuang. Secure or insure?: a game-theoretic analysis of information security games. In *WWW*, pages 209–218, 2008.

[16] H. Hu, G.-J. Ahn, and J. Jorgensen. Detecting and resolving privacy conflicts for collaborative data sharing in online social networks. In *ACSAC*, pages 103–112, 2011.

[17] H. Hu, G.-J. Ahn, and J. Jorgensen. Multiparty access control for online social networks: model and mechanisms. *IEEE Transactions on Knowledge and Data Engineering*, 25(7):1614–1627, 2013.

[18] P. Ilia, B. Carminati, E. Ferrari, P. Fragopoulou, and S. Ioannidis. Sampac: socially-aware collaborative multi-party access control. In *CODASPY*, pages 71–82, 2017.

[19] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

[20] U. Kuter and J. Golbeck. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In *AAAI*, volume 7, pages 1377–1382, 2007.

[21] M. Lesani and S. Bagheri. Applying and inferring fuzzy trust in semantic web social networks. In *Canadian Semantic Web*, pages 23–43, 2006.

[22] F. Paci, A. Squicciarini, and N. Zannone. Survey on access control for community-centered collaborative systems. *ACM Computing Surveys (CSUR)*, 51(1):6:1–6:38, Jan. 2018.

[23] S. Rajtmajer, A. Squicciarini, C. Griffin, S. Karumanchi, and A. Tyagi. Constrained social-energy minimization for multi-party sharing in online social networks. In *AAMAS*, pages 680–688, 2016.

[24] W. Sherchan, S. Nepal, and C. Paris. A survey of trust in social networks. *ACM Computing Surveys (CSUR)*, 45(4):47, 2013.

[25] A. Squicciarini, S. Rajtmajer, and N. Zannone. Multi-party access control: Requirements, state of the art and open challenges. In *SACMAT*, 6 2018.

[26] A. C. Squicciarini, M. Shehab, and F. Paci. Collective privacy management in social networks. In *WWW*, pages 521–530, 2009.

[27] A. C. Squicciarini, M. Shehab, and J. Wede. Privacy policies for shared content in social network sites. *The VLDB Journal*, 19(6):777–796, 2010.

[28] J. M. Such and N. Criado. Resolving multi-party privacy conflicts in social media. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1851–1863, 2016.

[29] J. M. Such and N. Criado. Multiparty privacy in social media. *Commun. ACM*, 61(8):74–81, 2018.

[30] J. M. Such, J. Porter, S. Preibusch, and A. N. Joinson. Photo privacy conflicts in social media: A large-scale empirical study. In *CHI*, pages 3821–3832, 2017.

[31] K. Thomas, C. Grier, and D. M. Nicol. unfriendly: Multi-party privacy risks in social networks. In *PETS*, pages 236–252, 2010.

[32] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011.

[33] N. Vishwamitra, Y. Li, K. Wang, H. Hu, K. Caine, and G.-J. Ahn. Towards pii-based multiparty access control for photo sharing in online social networks. In *SACMAT*, pages 155–166, 2017.

[34] R. Wishart, D. Corapi, S. Marinovic, and M. Sloman. Collaborative privacy policy authoring in a social networking context. In *POLICY*, pages 1–8, 2010.

[35] L. Xu, C. Jiang, N. He, Z. Han, and A. Benslimane. Trust-based collaborative privacy management in online social networks. *Forensics and Security*, 2018.

Appendix

Appendix A

For completeness, we included the results of the experiments presented in Section 3 for the remaining cases in the Viewing algorithm, that is, when the originator/contributor are not directly connected to the owner, i.e., distance ≥2, and the viewer is in the *PER* set (see Figure 1.7) and when she is not (see Figure 1.8).

Regarding the Sharing algorithm, we generated the combinations for a contributor when the distance ≥2 and the viewer is in the `permit_sharing` set (see Figure 1.9). Additionally, we run the experiments for the originator when the viewer is in the `permit_sharing` set. We generated two figures according to the trust level, i.e., when the *TG.infer* (*originator*,*owner*) returns 0.25 (see Figure 1.10a) and when it returns 0.75 (see Figure 1.10b).

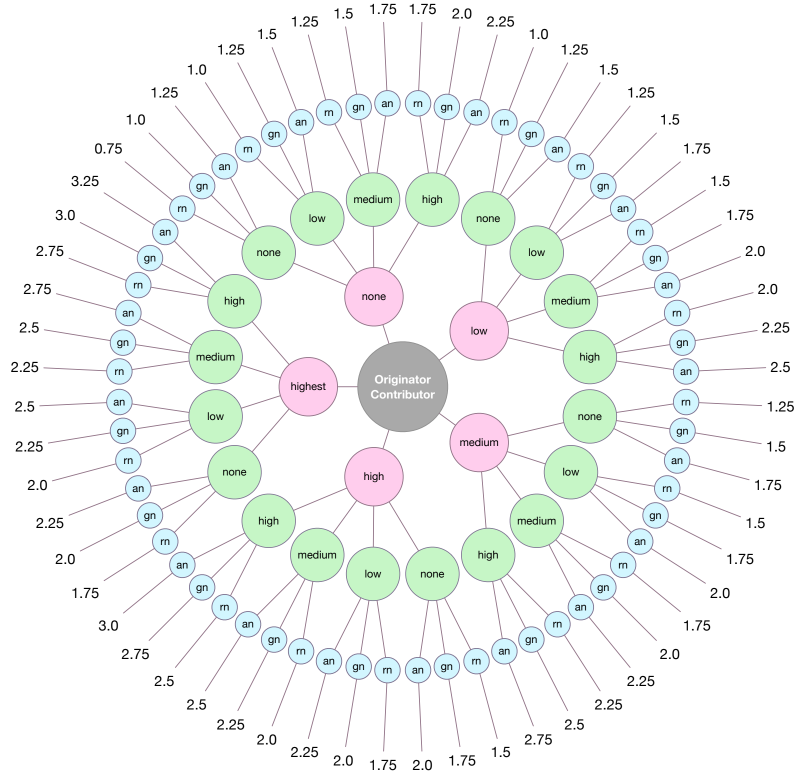


Figure 1.7: Viewing: Decision values for the contributor/originator with (ϕ_{ct}, ϕ_{at}, ϕ_{tr}, ϕ_{sl}) = 1; accessor ∈ *PER*; and distance≥2

Appendix B

Table 1.11: Viewing baseline probability of revoking the owner's initial decision for one contributor/originator when distance is ≥ 2

	Viewing — Contributor/Originator													
Output	4	3.75	3.5	3.25	3.0	2.75	2.5	2.25	2.0	1.75	1.5	1.25	1.0	0.75
Frequency	1	2	4	6	9	10	10	8	6	3	1	0	0	0
Revocation number	0	0	0	0	1	3	7	13	22	32	42	50	56	59
Baseline Probability	0%	0%	0%	0%	1.6%	5%	11.6%	21.6%	36.6%	53.3%	70%	0%	0%	0%

We executed the experiments in order to compute all the baseline probabilities for the Viewing algorithm when the distance of both the contributor and the originator is ≥2 (see Table 1.11).

For the Sharing algorithm, we included tables corresponding to the con-

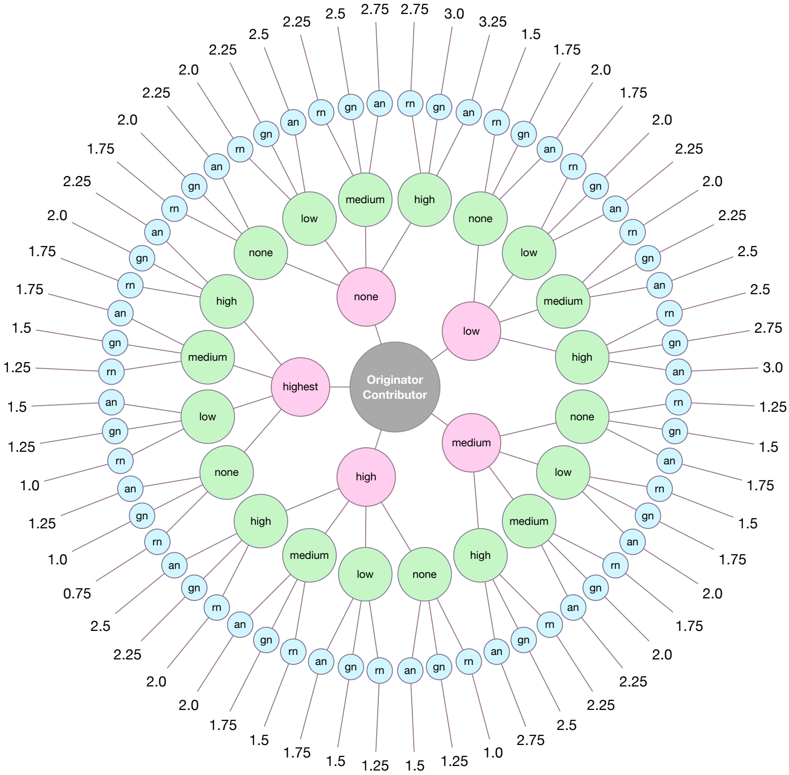


Figure 1.8: Viewing: Decision values for the contributor/originator with (ϕ_{ct}, ϕ_{at}, ϕ_{tr}, ϕ_{sl}) = 1; accessor ∈ *DEN*; and distance ≥2

tributor when the distance ≥2 (see Table 1.12) and to the originator when the weight is 0.25 (see Table 1.13).

Table 1.12: Sharing baseline probability of revoking the owner's initial decision for one contributor when distance ≥ 2

Sharing — Contributor						
Output	2	1.5	1.25	1.0	0.75	0.5
Frequency	1	1	1	1	0	0
Revocation number	0	0	0	1	2	3
Baseline Probability	0%	0%	0%	25%	0%	0%

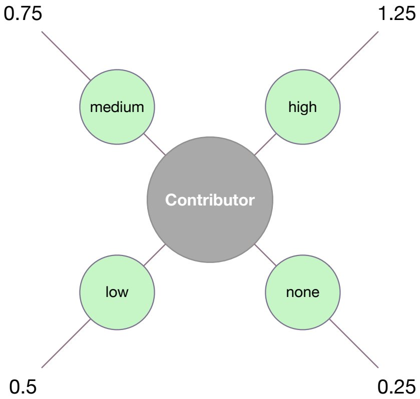
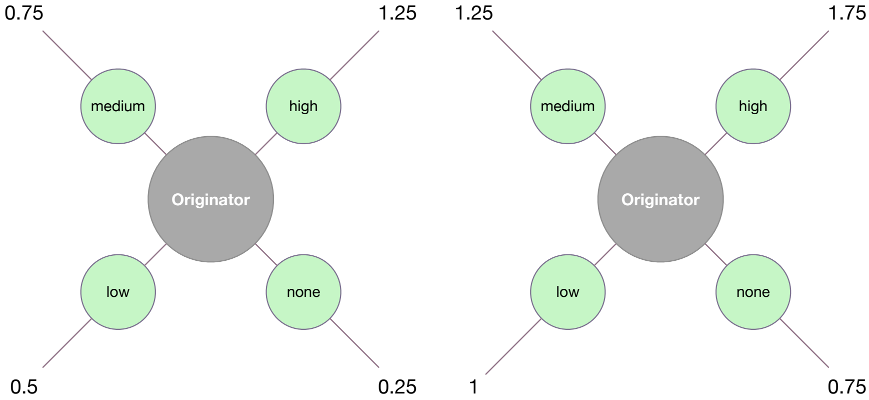


Figure 1.9: Sharing: Decision values for the contributor with $(\phi_{ct}, \phi_{sl}) = 1$; viewer $\in$ `permit_sharing`; and distance ≥ 2



(a) *TG.infer* (*originator*,*owner*)=0.25 **(b)** *TG.infer* (*originator*,*owner*)=0.75

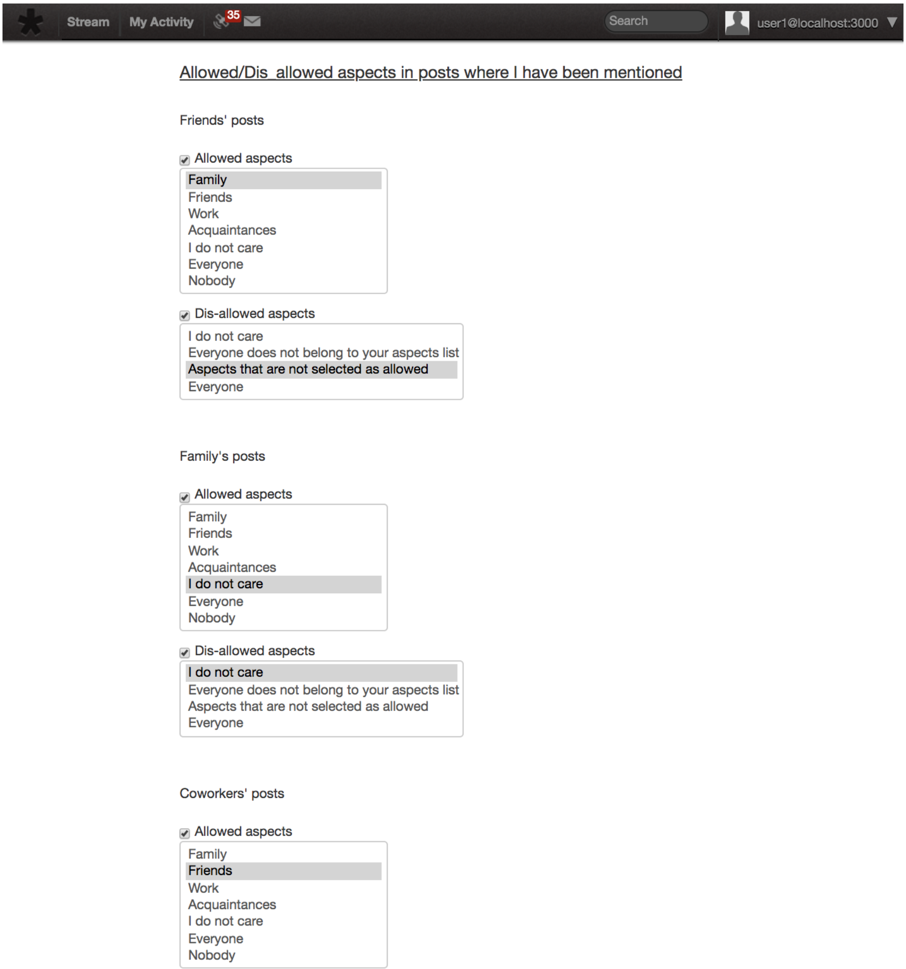
Figure 1.10: Sharing: Decision values for the originator with $(\phi_{ct}, \phi_{sl}) = 1$; viewer $\in$ `permit_sharing`

Appendix C

Here we show different screenshots of the UI of the collaborative access control prototype that we implemented in Diaspora. In Figure 1.11, associated controllers can specify their privacy preferences regarding who—accessors—are allowed to access the item and who are not. Figure 1.12 shows the settings of the sensitivity level, trust level and sharing policy that associated controllers can assign.

Table 1.13: Sharing baseline probability of revoking the owner's initial decision for one originator when her weight is 0.25

Sharing — Originator							
Output	2	1.5	1.25	1.0	0.75	0.5	0.25
Frequency	1	1	1	1	0	0	0
Revocation number	0	0	0	1	1	2	3
Baseline Probability	0%	0%	0%	25%	0%	0%	0%



Allowed/Dis allowed aspects in posts where I have been mentioned

Friends' posts

☑ Allowed aspects
- [✓ Family]
- Friends
- Work
- Acquaintances
- I do not care
- Everyone
- Nobody

☑ Dis-allowed aspects
- I do not care
- Everyone does not belong to your aspects list
- [✓ Aspects that are not selected as allowed]
- Everyone

Family's posts

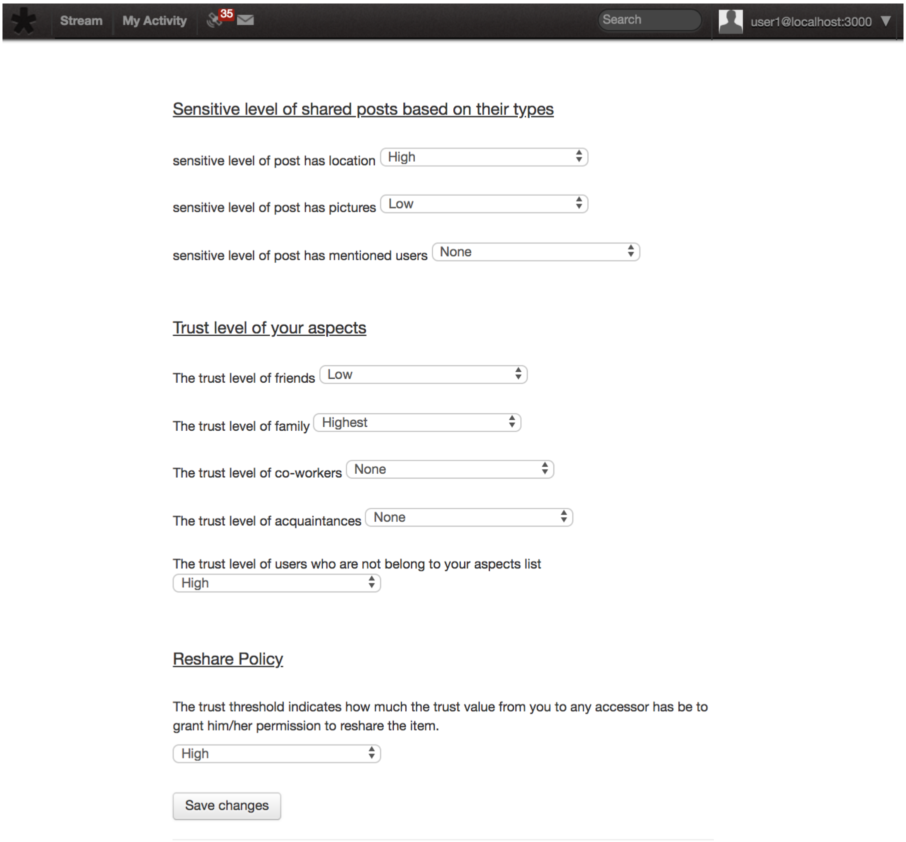
☑ Allowed aspects
- Family
- Friends
- Work
- Acquaintances
- [✓ I do not care]
- Everyone
- Nobody

☑ Dis-allowed aspects
- [✓ I do not care]
- Everyone does not belong to your aspects list
- Aspects that are not selected as allowed
- Everyone

Coworkers' posts

☑ Allowed aspects
- Family
- [✓ Friends]
- Work
- Acquaintances
- I do not care
- Everyone
- Nobody

Figure 1.11: User interface to specify allowed and disallowed users



Sensitive level of shared posts based on their types

sensitive level of post has location: High

sensitive level of post has pictures: Low

sensitive level of post has mentioned users: None

Trust level of your aspects

The trust level of friends: Low

The trust level of family: Highest

The trust level of co-workers: None

The trust level of acquaintances: None

The trust level of users who are not belong to your aspects list: High

Reshare Policy

The trust threshold indicates how much the trust value from you to any accessor has be to grant him/her permission to reshare the item.

High

Save changes

Figure 1.12: User interface to assign sensitivity levels to types of items and trust levels on other users