Solving Access Control Conflicts in Multi-User Systems

Alba Martinez Anton, Clara Bertolissi and Jean-Marc Talbot
LIS, UMR 7020 Aix-Marseille Université, CNRS, France

Keywords: Multi-Party Access Control, Policy Conflict Resolution, Provenance, Data Sharing, Social Networks.

Abstract: Collaborative systems deal with shared content which is jointly owned and managed by multiple users. Individual privacy preferences should be taken into account during access control evaluation, resolving conflicts while ensuring the acceptability of collective access decisions. In this work we propose a threshold-based conflict resolution strategy in the context of social networks. The resolution method is based on the information captured in the social graph, supporting the interpersonal relations between users, and the provenance graph, supporting the multi-management of data. A prototype implementation attests the feasibility of the proposed approach.

1 INTRODUCTION

In the past decades, because of the democratization of the new technologies, people has found new ways to keep in touch with each other or meet new people. Systems such as Online Social Networks (OSN) are mainly aimed for information sharing between their users, which has a positive impact for social interactions (Krasnova et al., 2010).

By nature, OSN platforms deal with data jointly owned by multiple users (hereafter called controllers). The controllers are often the host (the user hosting the object in his personal space), the owner (the user posting or uploading the object), and the stakeholders (all the other users involved in the object, e.g. users appearing on a picture). Controllers may have a disagreement on some access request generating a conflict that needs to be solved by finding a collective access decision. However, the classical conflict resolution algorithms (such as grant-override or deny-override) lack of granularity: they are either too prohibitive or, on the contrary, too permissive. Several conflict resolution methods have been proposed in the literature. For instance, several works propose the definition of negotiation protocols between controllers, other works apply theoretical models from game theory. However, such methods create an overload to the users, a delay for the requester and a loss of dynamics in the system.

We propose a new approach for collaborative access control for OSN by relying on two graph models: the social graph, describing the interpersonal relations between users in the system and the provenance graph, capturing the interactions between users and objects, namely the user requesting an access, the requested object and its controllers. We design a fine-grained threshold-based conflict management method, and removing as much burden as possible for the users. We consider some decision making factors from the literature, such as the sensitivity level of the requested object, or the interest of giving access to the requester, and we automatize their computation using information from the social and the provenance graphs. We also define new decision making factors, such as the information spread in a community of users, or the reliability of a piece of data. All these factors are combined together in a final conflict resolution function that has been experimentally tested on several scenarii to attest its practical relevance.

In the next sections, we present the related works (Sec. 2), we define the social network graphs in Sec. 3 and we present our conflict resolution model in Sec. 4. Finally, in Sec. 5, we briefly present an experimental validation of our model and we conclude in Sec. 6.

2 RELATED WORK

Conflict management in OSN has been widely studied over last years (Paci et al., 2018). Finding an optimal resolution algorithm for social networks is nontrivial and several solutions have been proposed. Due to the spread of XACML architectures (OASIS, 2013), the standard combining algorithms such as permit-override, deny-override or first applicable are well

Martinez Anton, A., Bertolissi, C. and Talbot, J.
Solving Access Control Conflicts in Multi-User Systems.
DOI: 10.5220/0012768500003767

In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 492-499
ISBN: 978-989-758-709-2; ISSN: 2184-7711

known, but they are not well-suited for conflict resolution in OSN. More fine-grained solutions use majority or consensus protocols (Alshareef et al., 2020; Carminati and Ferrari, 2011) or more complex strategies, like (Hu et al., 2013) where majority or consensus is applied after a vote based on weighted preferences and trust levels of the controllers. Other works apply game theory notions to negotiation for conflict resolution (Rajtmajer et al., 2016). However these methods have limitations when applied to OSN. Negotiation to agree on a decision creates additional burden for users. Game theory is not flexible enough to represent human behaviour due to social idiosyncrasies (Such and Criado, 2015), even if some works base their games on trade-off between privacy and utility (Hu et al., 2014).

Some works focus on access conflict management for pictures, proposing solutions to blur faces (Ilia et al., 2015) or parts of the picture (Vishwamitra et al., 2017). However, the generalisation of these methods to other type of content is not easy.

Other works base their resolution method on threshold values. These values are generally based on trust levels assigned to users or on sensitivity levels associated with objects. For example, in (Hu et al., 2013), the authors propose, in addition to the previously cited vote strategy, to use the computed decision value as a threshold to authorize or not the access. In (Such and Criado, 2015), the authors propose to estimate the willingness of changing their decision for the controllers, to be used as a threshold to modify the controllers decision. These methods may overload users with value calculation, often requiring manual parameter setting, which can be nontrivial. An alternative solution is the assistant ELVIRA (Mosca and Such, 2022) which implements a negotiation protocol without burdening the controllers. However the negotiation is done at the level of sets of users, loosing granularity.

Most of the methods in the literature are based on some common decision-making factors we briefly describe next. Some of them are used also in the solution we propose (see Sec. 4.1).

Trust: trust can be defined as the belief that a person is acting sincerely and won't do any harm. In conflict resolution methods, the trust or tie strength (Such and Criado, 2015) represents the degree of trust a user has in another one. Quantifying this concept between users unknown from each other is difficult and has lead to the design of dedicated algorithms for OSNs, such as TidalTrust (Golbeck 2005).

Privacy Preferences: users may have different points of view about personal information protection, usually reflected in their default policies. The privacy preferences give an estimation of the level of importance that privacy has for a user (Hu et al., 2011).

Item Sensitivity: the degree of sensitivity of an object is linked to the concept of privacy. The item sensitivity for a controller gives an indication of the harm that can be done to him if the object falls in wrong hands. This value can be an arbitrary value, only depending on the personal point of view (Hu et al., 2011) or it can be computed from the policies, relations and trust among users (Such and Criado, 2015).

Importance of a Conflict: the importance of a conflict measures the importance for a user that the conflict resolves conforming with his own preferences. This notion is used with the item sensitivity in (Such and Criado, 2015) to compute the willingness of a user to change his decision.

Sharing Gain vs Privacy Risk: the privacy risk estimates the risk of being harmed if the object is accessed by some requesters. It is usually used to balance the sharing gain, also called share loss (Hu et al., 2011) or sharing utility (Mosca and Such, 2022). The sharing gain measures the social advantages, such as maintaining relationships or mutual empathy (Krasnova et al., 2010), a user can have by giving access to an object.

3 OSN REPRESENTATION

Our work relies on threshold-based strategies based on some decision-making factors. These latter will be computed automatically from the social network model, composed of two graphs representing the interpersonal relations between users and the multi-management of data, respectively.

A social network is basically composed of a set of interconnected users and their shared content. Users in OSN have a personal space (profile page, wall,...) where they can share information such as texts, photos or videos. Users are connected to each other through interpersonal relationships, as for instance follower, friend, close friend, etc. Relationships may be asymmetric: for example, a user can follow someone, but not be followed back. Users have privacy preferences on what the other users can do with their information and on their personal spaces. These preferences are formalized as policies and are used to evaluate access requests. We denote $Pol_u(u',o)$ the evaluation result of the policy of the user u for the object o when the requester is u'. This expression can be evaluated to 0 (or 1), meaning the access is denied (or permitted) following this policy.

User relationships at a given time t are represented by a labelled directed graph $S = (\mathcal{U}, \mathcal{E}_s, L_s)$, where $\mathcal{U}$ are the nodes, $\mathcal{E}_s$ the edges and L_s a set of labels. Nodes represent users and a directed edge, labeled $\mathsf{isRelated}$, exists between two nodes if the first one is

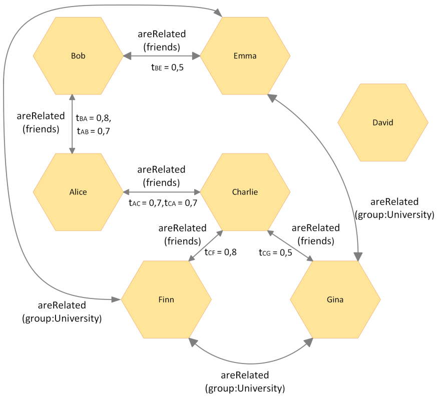


Figure 1: Social graph.

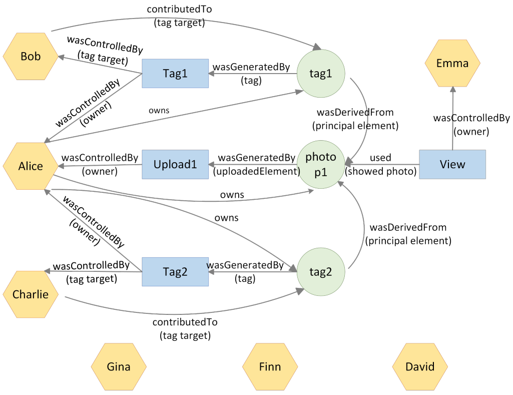


Figure 2: Provenance graph applied to social networks.

in a relation with the second one. The nature of the relation, e.g. *friends*, is specified as a role included in the edge label. We denote by areRelated the symmetric closure of isRelated.

Edges labeled with *friends* are annotated by a trust value $t \in [0,1]$ given by each user to his friends (see Sec. 4.1 for generalized trust value computation). Fig. 1 shows an example of a social graph (only the trust value of controllers is indicated to ease readability). Relationships can be used to define sets of users, called communities.

Definition 1. *(Community) Let rel be a binary relation between users and $\mathcal{U}$ be the set of users. The* ***group community*** *associated with rel is defined as $Com(rel) = \{u, u' \in \mathcal{U} \mid rel(u,u')\}$.*
A ***user-centered community*** *of radius k is defined as $Com_u(rel,k) = \{u' \in \mathcal{U} \mid rel(u,u')$ and $|u,u'| \leq k\}$ where $|u,u'|$ is the length of the path between u and u'.*

In our setting, *rel* represents a path in the social graph composed of isrelated edges. For example, referring to Figure 1, the Charlie-centered community of radius 1 built using the relation isRelated$(friends)$, is defined by $Com_{Charlie}(friends, 1) = \{Alice, Finn, Gina\}$.

As communities imply intuitively proximity among its users, it is reasonable to set an upper bound on the length of the relation.This bound is specific to the system and denoted *maxDis*.

To keep track of the object evolution and the links they share, we use the Open Provenance Model (Moreau et al., 2011). The information is represented as a labelled directed graph $P = (\mathcal{N}, \mathcal{E}_p, L_p)$, with $\mathcal{N}$ a set of nodes, $\mathcal{E}_p$ is set of directed edges, and L_p a set of labels. We denote $O, \mathcal{U}, \mathcal{P} \subset \mathcal{N}$ the sets of objects, users and processes. Two nodes are connected by an edge if there is a casual dependence between them. There are three basic dependencies: used which connects a process with an object used during its completion, wasGeneratedBy connecting an object with the process which created it, and wasControlledBy showing the interaction between a process and user. The OPM also contains composed casual dependencies, as wasDerivedFrom relating an object to the objects used to create it. Dependencies may include a role from the label set, providing a more precise indication of user or object roles in the process. An example of graph is shown in Fig. 2. The user nodes are depicted by hexagons, the processes by rectangles and the object by disks.

In addition to the standard model, we consider two casual dependencies presented in (Bertolissi et al., 2023): owns and contributedTo. The first one connects a user with an object if the user was controlling as owner the process which led to the object. The second one shows the dependence between a user and a object when the user was involved in the process leading to the object.

In the context of OSN, we may use such dependencies to retrieve the controllers of a given object. For instance, let us consider a photo with two types of controllers c: the owner and the tagged users. We can define the predicate controller for a photo as controller$(c, photo) = ($owns$(c, photo)) \vee ($contributedTo$(c, photo, "tag\ target"))$. For photo $p1$ in Fig. 2, we obtain Alice, as *owner*, and Charlie and Bob, as *tag targets*.

4 CONFLICT RESOLUTION MODEL

We propose a fine-grained conflict resolution method using the decision-making factors introduced in Sec. 2. More precisely, we propose to use the provenance graph (the history of the system) and the social graph (relations between users at the moment of the access) to quantify the *sensitivity* of an object and the *interest of sharing* it. Finally, we present a threshold conflict resolution formula based on the (weighted) ratio of the values associated with both sensitivity and the interest of sharing.

4.1 Sensitivity

The sensitivity is determined by the trust controllers have in users who may access the object, as well as the residual sensitivity from prior objects, calles inherited sensitivity.

User Trust in a Community. First of all, we need a trust value for each pair of users in the network. We already have in our model trust values assigned by any user to his friends. We propagate these values by using the Tidal Trust algorithm (Golbeck 2005) to automatically compute a trust value between any two connected users.

Definition 2 (User trust)**.** *To compute the trust between any two connected users at any distance, the Tidal Trust algorithm finds all the shortest paths between two users and compute the trust by applying the function* $\mathsf{Utrust} : \mathcal{U} \times \mathcal{U} \rightarrow [0,1] \subset \mathbb{R}$*:*

$$\frac{\sum_{j \in R \mid t_{(u_1,j)} \geq max} t_{(u_1,j)} \times \mathsf{Utrust}(j,u_2)}{\sum_{j \in R \mid t_{(u_1,j)} \geq max} t_{(u_1,j)}}$$

where $t_{(u,u')} \in [0,1]$ represents the trust between two users connected as friends at distance 1 in the social graph (see Sec. 3). The final trust value is computed by selecting relevant shortest paths using the max condition . We set the value max to 0.1 to explore multiple paths while maintaining trust estimation relevant (trust estimation in a third person is not significant if the trust between the first two is low).

Example 1. *Based on the scenario in Fig. 1, in Table 1 we report the (underlined) trust values between a controller and his friends. The remaining trust values are computed using the function* Utrust*. If we focus on Alice, the trust values for David, Emma, Finn and Gina need to be computed. Only one shortest (friends) path exists from any user to any another one in this scenario.*

$$\mathsf{Utrust}(A,E) = \frac{t_{(A,B)} \times \mathsf{Utrust}(B,E)}{t_{(A,B)}} = \frac{0.7 \times 0.5}{0.7} = 0.5$$

Here $\mathsf{Utrust}(B,E) = t_{(B,E)}$ *because Bob and Emma are friends. Similarly,* $\mathsf{Utrust}(A,F) = 0.8$ *and* $\mathsf{Utrust}(B,F) = 0.8$.

Using the trust between any two users in the system, we define the trust of a controller in his controller-centered community for an object, inspired from (Such and Criado, 2015).

Definition 3 (Trust in a community)**.** *The trust of a controller c in his controller-centered community of radius k for an object o is defined as* $\mathsf{Ctrust} : \mathcal{R}el \times \mathbb{N} \times \mathcal{U} \times \mathcal{O} \rightarrow [0,1] \subset \mathbb{R}$*, where $\mathcal{R}el$ is the set of community relations, We have* $\mathsf{Ctrust}(path,k,c,o) =$

$$\min_{u \in Com_c(path,k)} (\mathsf{Utrust}(c,u) \times Pol_c(u,o))$$

The relation path refers to the relation building the community and is system-specific.

Table 1: Trust between users.

	A	**B**	**C**	**D**	**E**	**F**	**G**
A	x	$\underline{0.7}$	$\underline{0.7}$	0	0.5	0.8	0.4
B	$\underline{0.8}$	x	0.7	0	$\underline{0.5}$	0.8	0.4
C	$\underline{0.7}$	0.7	x	0	0.5	$\underline{0.8}$	$\underline{0.4}$

As this value is used to compute the sensibility, considering the trust of the less trustworthy user reflects the level trust needed to access the object. In our running scenario, we consider as *path* the relation *friends* and we fix the user-centered community radius to $k = 1$.

Example 2. *Let us consider photo $p1$ in Fig.2, its three controllers Alice, Bob and Charlie and their user policies as follows:*

$$\begin{aligned} pol(A,p1) &= \textit{friends OR friends of friends} \\ pol(B,p1) &= \textit{friends OR Charlie} \\ pol(C,p1) &= \textit{all EXCEPT Gina} \end{aligned}$$

Within a community only the trust of a controller c in the users that c allows to access the object is taken into account. For example the community trust of Charlie will not use the trust that Charlie has in Gina, but only the ones in Alice and Finn. According to the scenario in Fig.1, we obtain $\mathsf{Ctrust}(friends,1,A,p1) = \mathsf{Ctrust}(friends,1,C,p1) = 0.7$ *for Alice and Charlie, and* $\mathsf{Ctrust}(friends,1,B,p1) = 0.5$ *for Bob.*

Sensitivity of an Object. The sensitivity of an object is related to its controllers, but it is independent from the requester. Intuitively, if a user with low trust is allowed by any of the controller's policies to access the object, then the object sensitivity should not be too high. To evaluate the sensitivity of an object o, we consider the trust in the community for the object o of each of its controllers, as well as the trust in the community for the objects from which o is derived.

For example, to find the sensitivity of a photo in a social network, the community trust of the different controllers of the photo is considered as well as those of the controllers of the album containing the photo (i.e. inherited sensitivity)

Definition 4 (Inherited sensitivity)**.** *The inherited sensitivity of an object o, denoted* $\mathsf{HSens} : \mathcal{O} \rightarrow [0,1] \subset \mathbb{R}$*, is defined as a combination of the controller-centered community trusts with respect to the objects from which the requested object o is derived:*

$$\mathsf{HSens}(o) = \underset{\substack{o' \neq o \mid \mathsf{wasDerivedFrom}^{+}(o,o') \\ c \in P \mid \mathsf{controller}(c,o'),}}{average} \mathsf{Ctrust}(path,k,o',c)$$

We also consider a variation HSens^+ of the inherited sensitivity function, where o is included in the set of derived objects, i.e. the constraint $o' \neq o$ is discarded.

The sensitivity of an object o is given by combining the values of the trust on the community of the different controllers of o, as well as the inherited sensitivity of the object. The aim is to mitigate the sensitivity of the object (obtained by combining the controller-centered community trusts) by using the ratio of the inherited sensitivity before and after the existence of the object.

Definition 5 (Sensitivity). *The sensitivity function* $\mathsf{Sens} : O \to [0,1] \subset \mathbb{R}$ *is defined as* $\mathsf{Sens}(o) =$*:*

$$\frac{\mathsf{HSens}(o)}{\mathsf{HSens}^+(o)} \times \underset{c \in P | \mathsf{controller}(c,o)}{average} \mathsf{Ctrust}(path, k, c, o)$$

Example 3. *Let us compute the sensitivity for photo* `p1` *wrt controllers Alice, Bob and Charlie, whose trust in their communities is computed in Ex.1. We have* $\mathsf{Sens}(p1) = average(0.7, 0.5, 0.7) = 0.63$*. Since the sensitivity value is in* $[0,1]$*, the sensitivity of p1 is medium-high. Notice that in the example there are no objects from which the photo is derived.*

4.2 Interest of Sharing

The interest of sharing, quantifying the benefit of sharing an information with a user, depends on the requested object and on the requester. We assume that the benefit increases when the piece of information is reliable and trustworthy (*accuracy*) and when the piece has not been widely disseminated (*community spread*). Both notions are computed from the controllers point of view.

Accuracy of an Object. The accuracy of an object measures its quality: the more the controllers trust each other, the higher the reliability of the object (from the controllers point of view) is. We define thus the accuracy of an object o for a controller c as the combination of the mutual trusts between c and the controllers of o and the objects from which o is derived.

Definition 6 (Accuracy for a controller). *The accuracy of an object o for a controller c,* $\mathsf{Acc}_c : O \times \mathcal{U} \to [0,1] \subset \mathbb{R}$*, is defined as the minimal trust of c in the controllers of the objects from which o is derived:*

$$\mathsf{CoAcc}(o,c) = \underset{\substack{o' | \mathsf{wasDerivedFrom}^+(o,o') \\ c' \in \mathcal{U} | \mathsf{controller}(c',o')}}{min} \mathsf{Utrust}(c,c')$$

The accuracy of an object is computed by combining the controllers individual accuracy values.

Definition 7 (Global accuracy). *The global accuracy of an object, denoted* $\mathsf{Acc} : O \to [0,1] \subset \mathbb{R}$*, is defined as:*

$$\mathsf{Acc}(o) = \underset{c \in \mathcal{U} | \mathsf{controller}(c,o)}{average} \mathsf{CoAcc}(o,c)$$

Example 4. *We want to determine the accuracy of photo p1, from Fig. 2. The object p1 has no objects from which it is derived. The controllers to be considered are Alice, Bob and Charlie. Using the values of trust in Ex. 1, the accuracy of* $p1$ *for controller U, with* $U \in \{Alice, Bob, Charlie\}$*, is* $\mathsf{CoAcc}(p1, U) = 0.7$*, and the accuracy for p1 is* $\mathsf{Acc}(p1) = average(0.7, 0.7, 0.7) = 0.7$.

Community Spread. The previous accesses to an object may influence the interest in further granting access to it. To quantify the notion of information spread inside a community, we retrieve the accesses *per community* that have been made to the requested object within a time interval (specific to each system): for every access request, we determine which community containing the requester has been granted with the greatest number of accesses. To determine those communities, we look at all the relations in the social graph of distance 1 from the requester. Accesses in the predefined time interval are grouped by community and the biggest number of accesses is chosen.

It's worth noting that the notion of community spread doesn't depend on a controller of the object, but on the object itself and on the requester.

Definition 8 (Community Spread of an object). *The community spread,* $\mathsf{Spread} : O \times \mathcal{U} \to \mathbb{R}$*, is defined by* $\mathsf{Spread}(o, u_r) =$

$$\begin{cases} 1\ if \underset{com \in Co_{u_r}}{max} \frac{ln(e + access(com,o))}{\lambda} < 1 \\ \underset{com \in Co_{u_r}}{max} \frac{ln(e + access(com,o))}{\lambda}\ otherwise \end{cases}$$

We denote u_r *the user making the request and by* $Co_{u_r} = \{com | u_r \in com\}$ *the set of communities to which the requester belongs. The term access(com,o) denotes the number of (granted) accesses made by the users of the community com to the object o. The coefficient* λ *is a positive constant (that may depend on the system) .*

Following the definition above, the more an object has been accessed by a community to which the requester belongs, the highest is its Spread value. The use of a logarithm factor as well as the λ factor mitigates the increase of the spread value, which is essential for computing a reasonable interest of sharing.

Example 5 (Community Spread). *Consider an access request to photo* $p1$ *made by Finn. The communities to consider are* $Co_F = \{Com_F(\mathsf{isRelated}(F, x, friends), 1), Com(\mathsf{isRelated}(F, y, group : University), 1)\}$*, according to Fig. 1. Recently, according to Fig. 2, only Emma, in the University group community, got access to p1, i.e.* $\mathsf{accesses}(com, p1) = 1$*. Let the constant* λ *be set* *to $\lambda = 1.7$ (see Sec.5 for details). Then, we obtain $\mathsf{Spread}(p1, Finn) = 1$, since $\frac{ln(e+1)}{1.7} < 1$. Since the number of accesses is very low, the spread of photo $p1$ will not impact its interest of sharing value (see Ex.6).*

Definition 9 (Interest of Sharing)**.** *The interest of sharing, $\mathsf{SInt} : O \times \mathcal{U} \to \mathbb{R}$, is defined by:*

$$\mathsf{SInt}(o, u_r) = \mathsf{Acc}(o) \times \frac{1}{\mathsf{Spread}(o, acc)}$$

The interest of sharing depends on the quality of the object itself and on the level of its spread in the communities to which the requester belongs.

According to the definition, a requester who joins a community where a piece of information has been largely spread, has a lower interest of sharing compared to the interest of sharing he had as an outsider.

Example 6 (Interest of Sharing)**.** *The interest of sharing photo p1 for user Finn is $\mathsf{SInt}(p1, F) = 0.7 \times \frac{1}{1} = 0.7$. The same computation applies to Gina. The controllers' interest of sharing with Finn or Gina is high, meaning the controllers will benefit from sharing the photo with them.*

4.3 Conflict Resolution Function

In order to compute a collective access decision when controllers' preferences differ, we present a conflict resolution strategy which uses the previously defined decision-making factors.

Intuitively if the sensitivity of the object is high and the interest of sharing is low, conflict resolving should return a deny decision. In the opposite situation, access to the object should be granted. However, the ratio between sensitivity and interest of sharing doesn't consider the trust the controllers have in the requester, which can give some indication about the harm or the benefit the access can cause. In our example, if we don't consider the controllers' trust, Finn and Gina would have both access to the photo. To solve this problem, we balance the sensitivity and the interest of sharing with two coefficients, denoted α and β, respectively.

The coefficient α represents the trust the controllers have in the requester, supposing they deny him the access to the requested object in their privacy preferences. If the controllers trust the requester highly, there's little risk of harm, even if the object is sensitive. Conversely, if they distrust the requester, the risk of harm increases. Following this idea, the α coefficient minimizes the sensitivity of the object. We define then α as

$$2 - \min_{c \in P | \mathsf{controller}(c,o)} (\mathsf{Utrust}(c, u_r)) \times (1 - Pol_c(u_r, o)))$$

The coefficient β is used to balance the interest of sharing and represents the trust the controllers have in the requester, supposing they give him the access to the object in their privacy preferences. Controllers are more willing to share the object if they trust the requester. If trust is low, they are less inclined to share. Hence,

$$\beta = 1 + \max_{c \in P | \mathsf{controller}(c,o)} (\mathsf{Utrust}(c, u_r) \times Pol_c(u_r, o))$$

Due to the definition of the function Utrust, the values of α and β are between 1 and 2.

Definition 10 (Conflict resolution)**.** *The final conflict resolution formula is given by :*

$$\mathsf{R(u_r, o)} = \frac{\alpha \times \mathsf{Sens(o)}}{\beta \times \mathsf{SInt(o, u_r)}}$$

with α and β the coefficients defined as above.

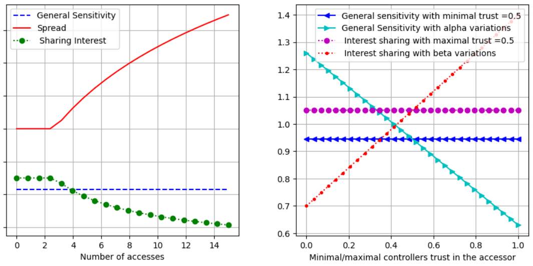
When $\mathsf{R(u_r, o)} \geq 1$, the sensitivity of the object is high in comparison to the interest of sharing to authorize the access, so the access decision will be *Deny*. On the contrary, when $\mathsf{R(u_r, o)} < 1$, the interest of sharing is more important than the sensitivity, so the access request will be *permitted*.

Example 7. *Consider again the scenario where Finn (or Gina) asks to access the photo $p1$.*

The trust the controllers put in Finn is 0.8 (see Ex. 1). The coefficients α and β are $\alpha = 1.2$, and $\beta = 1.8$. The conflict resolution, $\mathsf{R(F, p1)} = 0.6 < 1$, grants the access for Finn (interest of sharing is higher than sensitivity and Finn is highly trusted by all controllers). Let us consider now the requester Gina, whose SInt and Sens are the same as Finn's. Only the α and β coefficients can make the conflict outcome different. The trust the controllers put into Gina is medium low, 0.4. The conflict resolution value is $\mathsf{R(Gina, p1)} = 1.03 > 1$, thus Gina will have her access to photo $p1$ denied. Due to a medium-low trust in the requester and the closeness between SInt and Sens, the weight of α coefficient leads to a negative response for Gina. If the two values had a significant difference, then α and β would not affect the conflict outcome.

5 DISCUSSION AND VALIDATION

The decisions-making factors we consider are based on the trust between users and their relation with the requested object. It is not difficult to suppose that an individual is more likely to let someone access a piece of information if the trust in him is high. Our model follows this assumption.



(a) Impact of the community spread on the interest of sharing

(b) Influence of α and β coefficients

Figure 3: Factors evolution.

As reported in Fig.3, when the sensitivity (*i.e.* the potential harm) increases, the chance the object to be accessed decreases: indeed, either the sensitivity becomes larger than the interest of sharing or the difference between them decreases. Hence, it will be more likely that due to the α coefficient, the value of the resolution formula will be smaller than 1. Conversely, when the interest of sharing (*i.e.* the utility in permitting the access) increases, the outcome of the conflict resolution is more likely to be positive (because the interest of sharing becomes larger than sensitivity with or without β). Remind that if the information has been widely spread, the interest of sharing becomes smaller (Fig. 3a). By the λ coefficient (set in our scenarios to 1.7), the community spread is regulated and has impact on the interest of sharing without totally overriding the accuracy. As shown in Fig. 3b, the high trust of the controllers in a requester has two consequences: firstly, the risk of harm being reduced, both the α coefficient and the sensitivity weighted by α decreases. Secondly,the interest of sharing value when weighted by the β coefficient is high.

Comparison to other Threshold-Based Models. As presented in Sec. 2, the most common idea in threshold based models is to balance privacy risks with benefits of sharing. Usually privacy risks are computed using the sensitivity of an object, the trust in the requester and the privacy preferences of the controllers. The first two factors are taken into account in our formula by the sensitivity and the α coefficient. Privacy preferences are used to balance the sensitivity value, because in the models where the value is given directly by the controllers, it can be exaggerated to induce a decision in their favor. In our case the value is computed from the social network platform, so there is no need of mitigation. The benefit of sharing is usually measured by computing a dual of privacy risk. In our case we use different values to provide a quantification.

In (Such and Criado, 2015) the willingness of change is computed. This model measures the importance of a conflict, based either on the trust on the community or the sensitivity of the object together with the trust in the requester. The greater is the difference between one of the first two values and the last one, the higher is the importance of the conflict. Our approach is different, but our model tend to coincide on the outcome. Following their scenario, our model leads to the same access decisions for small community spread values. By making vary the parameter λ, we can force the model to obtain the same output for larger community spread.

Prototype Results. To validate our approach, we have developed a generator of graphs simulating a social network system, and of users policies. The social graph edges are generated following the friend to friend model (Levens et al., 2022). For each user, social content (e.g. photos, posts) is added to the provenance graph. Concerning user policies, based on Facebook options, we create for each user a file containing his preferences for the actions *view*, *post*, *share*, *comment* for the different types of objects (*photo, post, album, profile*). To compare our model with the other conflict resolution strategies, we have run 100 access queries on 10 different pairs of provenance and social graphs using our conflict resolution method, deny-override (DOV), permit-override (POV) and host-override (HOV) (Facebook current method).

We observe that our solution is more balanced in terms of denies and permits than POV and DOV algorithms. The POV algorithm resulted in 97% Permit decisions, while the DOV algorithm computed 25% Permits. Using our resolution method, the number of Permits was in average 32%. We obtain a slightly higher proportion of positive decisions than with the HOV method (40% of Permits). However, in average 21% of the requests have a different answer when our strategy is used instead of HOV. This is explained by the fact that in HOV permit decisions are based on the host preferences. Our model is more fine-grained, in the sense that the sensitivity of the object or the interest of sharing can lead to a denial of access, independently of the host preferences.

6 CONCLUSION

We have presented a fine-grained collaborative decision making model for social networks, using automatically computed decision making factors quantifying relevant aspects of the requested object and of the relations between controllers and requester, yielding a threshold formula. A proof-of-concept prototype is used to evaluate our model w.r.t. other conflict resolution methods. A validation through a user study is left for future work.

ACKNOWLEDGEMENTS

This work is partially funded by the Excellence Initiative of Aix-Marseille - A*MIDEX(Archimedes Institute AMX-19-IET-009), a French ”Investissements d’Avenir” Programme.

REFERENCES

Alshareef, H., Pardo, R., Schneider, G., and Picazo-Sanchez, P. (2020). A collaborative access control framework for online social networks. *J. Log. Algebr. Methods Program.*, 114:100562.

Bertolissi, C., Martinez Anton, A., and Zannone, N. (2023). Data sharing in social networks. In *Proc.* SACMAT ’23, page 181–192. ACM.

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

Golbeck, J. A. (2005). *Computing and Applying Trust in Web-Based Social Networks*. PhD thesis.

Hu, H., Ahn, G.-J., and Jorgensen, J. (2011). Detecting and resolving privacy conflicts for collaborative data sharing in online social networks. In *Proc.*, ACSAC ’11, page 103–112. ACM.

Hu, H., Ahn, G.-J., and Jorgensen, J. (2013). Multiparty access control for online social networks: Model and mechanisms. *IEEE Trans. Knowl. Data Eng.*, 25(7):1614–1627.

Hu, H., Ahn, G.-J., Zhao, Z., and Yang, D. (2014). Game theoretic analysis of multiparty access control in online social networks. In *SACMAT*.

Ilia, P., Polakis, I., Athanasopoulos, E., Maggi, F., and Ioannidis, S. (2015). Face/off: Preventing privacy leakage from photos in social networks. In *Proc. SIGSAC*, CCS ’15, page 781–792. ACM.

Krasnova, H., Spiekermann, S., Koroleva, K., and Hildebrand, T. (2010). Online social networks: Why we disclose. *J. Inf. Technol.*, 25(2):109–125.

Levens, W., Szorkovszky, A., and Sumpter, D. J. T. (2022). Friend of a friend models of network growth. *Royal Society open science*, 9(10).

Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., and den Bussche, J. V. (2011). The Open Provenance Model core specification (v1.1). *FGCS*, 27(6):743–756.

Mosca, F. and Such, J. (2022). An explainable assistant for multiuser privacy. *AUTON AGENT MULTI-AG Journal*, 36(1).

OASIS (2013). eXtensible Access Control Markup Language (XACML) VER 3.0. OASIS Standard.

Paci, F., Squicciarini, A., and Zannone, N. (2018). Survey on access control for community-centered collaborative systems. *ACM Comput. Surv.*, 51(1).

Rajtmajer, S., Squicciarini, A., Griffin, C., Karumanchi, S., and Tyagi, A. (2016). Constrained social-energy minimization for multi-party sharing in online social networks. In *Proc.* AAMAS ’16, page 680–688. SC. IFAAMAS.

Such, J. M. and Criado, N. (2015). Resolving multi-party privacy conflicts in social media. *IEEE Trans. Knowl. Data Eng.*, 28:1851–1863.

Vishwamitra, N., Li, Y., Wang, K., Hu, H., Caine, K., and Ahn, G.-J. (2017). Towards pii-based multiparty access control for photo sharing in online social networks. In *Proc.* SACMAT ’17, page 155–166. ACM.