# A Privacy Model for Social Networks

Alban Gabillon

# A Privacy Model for Social Networks

Alban Gabillon

Université de la Polynésie Française, Laboratoire GePaSud
BP 6570, 98702 FAA'A, Tahiti, Polynésie Française
alban.gabillon@upf.pf

**Abstract.** This paper defines a new multilevel privacy model for social networks like Facebook. This model is user-friendly i.e. it does not require the users to alter some security settings. It provides the users with a privacy policy with a high expressive power. First, authorizations are based on the type of relationships that the users have between them. Second, relationships themselves are protected.

## 1 Introduction

Access control solutions offered by existing social networks are not satisfactory. For example, in [1], the authors underline the fact that most Facebook users never alter their default privacy settings. The main reason is that most of the many Facebook privacy settings cannot be understood by users who are not security specialists. Moreover, most of the existing solutions lack flexibility and does not provide users with expressive privacy policies. In particular authors in [2] observe that most solutions do not take into account the type of relationships among users. In this paper, we define a new privacy model for social networks based on the multilevel security paradigm. The privacy policy is *mandatory* i.e. it applies to all users and cannot be updated. This means that, *there are no settings that the users should alter*. Basically, the only security related task that users should do is to categorize their contacts according to the relationship they have with them (friend, colleague, family etc.). Moreover, our model provides a privacy policy with a high expressive power. First, privileges of users depend on the type of relationship they have with the node they are browsing. Second, sensitive relationships are themselves protected. Based on our model, we also sketch an administration tool which automatically derives from the relationships, the Facebook security settings implementing the multilevel privacy policy.

In section 2, we briefly recall the multilevel security paradigm. In Section 3, we review the existing Facebook model. In Section 4 we define an unprotected social network. In Section 5 we define a multilevel social network. Section 6 is about security administration. Section 7 defines the mandatory privacy policy. In Section 8, we present our administration tool for the Facebook network. Section 9 reviews related work. Finally, Section 10 concludes this paper.

## 2 Multilevel Security

Multilevel security can be defined as follows: in Multilevel Security, a *security level* is assigned to each subject and each object. Security levels form a *lattice* (i.e. a partially ordered set with a lower bound and an upper bound). The security policy applies to all users and cannot be modified. If the objective is to guarantee the data confidentiality then the security policy (referred to as the *multilevel security policy*) can be expressed as follows: "subjects at a given security level are permitted to know everything about the data at the same or a lower level but are forbidden to know anything about the data at a higher or incomparable level". Bell & La Padula [3] showed that it is necessary to enforce the following two access control properties in order to guarantee the confidentiality policy:

- The *No-Read-up* property states that a subject at a given level of confidentiality cannot read an object at a higher or incomparable level.
- The *No-Write-down* property states that a subject at a given level of confidentiality cannot write to an object at a lower or incomparable level. The No-Write-down restriction is necessary to prevent a Trojan horse which runs on behalf of a high level user, from copying high level data into a low level object.

When it is assigned to an object the security level is referred to as a *classification level*. When it is assigned to a subject, the security level is referred to as a *clearance level*. A user with a high clearance level can initiate a session at a *working level* that is lower than or equal to his/her clearance level. This allows high-level users to update low-level objects without violating the No-Write-down restriction. Administrating the security policy in mandatory access control models means assigning security levels to subjects and objects. This is usually done under the responsibility of a single security officer.

## 3 The Facebook Privacy Model

In [4], the authors analyze the Facebook privacy model. They put in evidence that each Facebook user *U* has to manage four types of policies:

- a *search policy* which specifies whether Facebook users can reach a search listing of user *U* by performing a global name search.
- a *traversal policy* which specifies whether users, who have reached the search listing of user *U*, can see the friends of user *U*.
- a *communication policy* which specifies whether users, who have reached the search listing of user *U*, can communicate with user *U*.
- an access *policy* which specifies whether users, who have reached the search listing of user *U*, can access data objects on user *U*'s node. Regarding the access policy, users may have to specify up to 25 settings !

The Facebook privacy model is basically a discretionary model where the owner of a node has to specify the various actions (see, post, comment, etc.) that subjects (everyone, friends of friends, friends and specific friends) can perform on objects (profile items, wall messages, photos etc.). Without entering into the details of this model, we can formulate two major criticisms:

- Configuring the access policy is very confusing. Predicting precisely the outcome of the access policy execution is nearly impossible for users who are not security specialists. This problem has been reported by many authors (see [1] for instance).
- The traversal policy is very limited. Organizing friends into *Facebook lists* allows Facebook users to express that some of their friends are permitted to see their entire global friends' list while the other friends are forbidden to see it. However, users cannot hide *some* of their friends from some of their other friends. This limitation has led many Facebook users to open up several Facebook accounts (one for the "real" friends, one for the family, one for the colleagues etc.)

The new privacy model we define in this paper solves these two major problems.

## 4 Social Network

A Social Network is an undirected graph with nodes containing data objects. For representing such a graph, we define the following predicates:

- $node(u)$ reads "$u$ is a node"
- Several predicates for representing the various data shared on nodes, e.g. $age(u,a)$ which reads "age of node $u$ is $a$"
- $contact(u,v)$ reads "node $u$ and node $v$ are contacts"

A social network $S = N \cup R$ consists of,

- a set of nodes $N$ representing users and their pages. In other words $N$ contains instances of the *node* and the various data predicates.
- A set of edges $R$ representing contacts between users. In other words $R$ contains instances of the *contact* predicate.

Figure 1 shows a small social network with four nodes and the contact relationships between them.

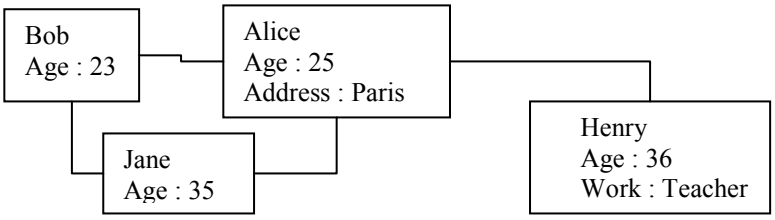


**Fig. 1.** Social Network.

The following set $S$ represents the social graph depicted in Figure 1:

$$\{node(Bob), contact(Bob, Alice), Age(Bob, 23), etc.\}$$

## 5 Multilevel Social Network

Each node can be seen as a small database that is administered by the owner of the node. Our aim in this paper is to design a multilevel privacy model for social network

where the only security related task assigned to users is to categorize the type of relationship they have with their contacts.

Our model is presented in an informal way for better readability. Let us however mention that we formalized it in first-order logic and implemented it as a Prolog [10] knowledge base proving its correctness and completeness.

### 5.1 Privacy Levels

*Privacy levels* represent the various relationships users can have between them. We assume that the set of level forms a lattice associated with the partial order relation *dominate* $\geq$ which is transitive, reflexive and anti-symmetric. Throughout this paper, we shall use the set of privacy levels depicted in Figure 2. Level *Foaf* dominates level *Everyone*. Level *Friend* dominates level *Foaf* etc. Levels *Family*, *Colleague* and *Friend* are incomparable. Levels *Family* and *CloseFriend* are incomparable.

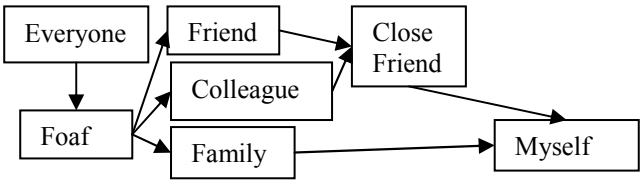


**Fig. 2.** Lattice of privacy levels.

### 5.2 Multilevel Nodes

We define the set of multilevel nodes $N^C$ as the set obtained by classifying the facts belonging to $N$. We define a multilevel Social Network as follows: $S^C = N^C \cup R$. The following Figure 3 depicts an example of multilevel social network.

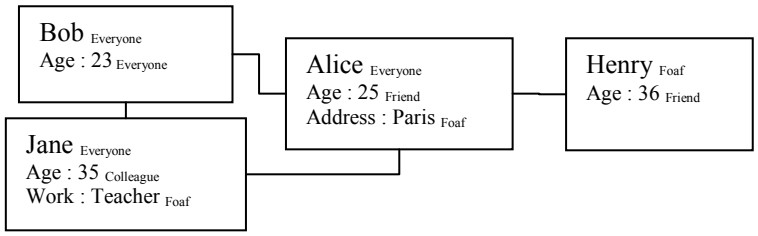


**Fig. 3.** Multilevel Social Network.

Note that only nodes and their content are classified. Contact edges between the nodes are not explicitly classified. This does not mean that everybody may see other people's contacts. We show in section 7.3 how we protect the existence of these relationships.

# 6 Security Administration

## 6.1 Clearance Levels

In a centralized multilevel database there is one security officer who assigns one clearance level to each user. In our model, users administrate their own node. The only task that users should do is to categorize their contacts according to the kind of relationship they have with them (friend, family, etc.). *This task corresponds to organizing contacts into lists in the current Facebook system,* with the major difference that Facebook lists do not form a lattice. In Figure 2, the levels which are strictly between the *Foaf* level and the *Myself* level represent the various kind of possible relationships.

Let us assume that each user categorized their contacts as follows:

- Bob's contacts: Alice (Friend), Jane (Friend)
- Jane's contacts: Bob (Close Friend), Alice (Colleague)
- Alice's contacts: Jane (Friend), Henry (Family), Bob (Friend)
- Henry's contacts: Alice (Family)

Figure 4 shows how Alice categorized her contacts.

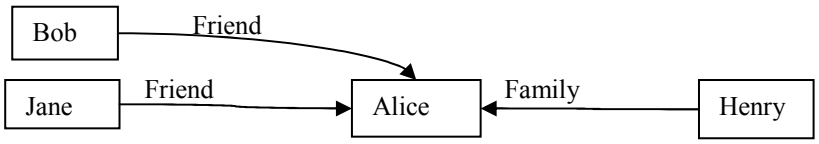


**Fig. 4.** Clearance levels of users browsing Alice's page.

Figure 5 shows how other users categorized their relationship with Alice.

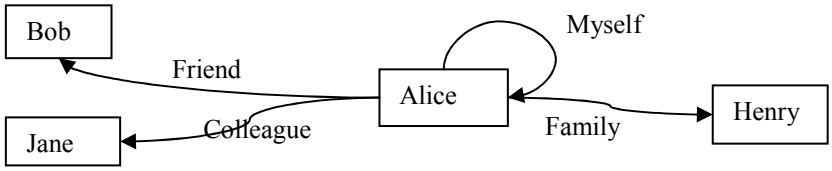


**Fig. 5.** Clearance levels of Alice.

Given a user *v* browsing a node *u*, the clearance level of user *v* is *dynamically* computed according to the four following rules:

1. If *u*=*v* (i.e. user *u* is browsing her own page) then the clearance level of user *u* is always *Myself*. For instance, the clearance level of Bob when accessing to his own node is *Myself*.
2. If user *v* is a contact of a contact of user *u* then the clearance level of user *v* is always *Foaf*. For instance, the clearance level of Henry when accessing to node Bob is *Foaf*.
3. If user *v* is neither a contact nor a contact of a contact of user *u* then the clearance level of node *v* is always *Everyone*.

4. If user *v* is a contact of user *u* and if user *u* categorized the relationship between her and user *v* as *l*, then the clearance level of user *v* is *l*. For instance the clearance level of Alice when accessing to node Jane is *Colleague*.

### 6.2 Working Level

Let *l* be the clearance level of user *v* when browsing node *u*. User *v* may set her working level to any privacy level *l'* provided level *l* dominates level *l'*. Very often, a user would not need to do any specific action to lower his/her working level. The default working level of a user is of course equal to her current clearance level which was assigned by the owner of the node that the user is browsing. If a user browsing a node needs to update a low level message (for adding a comment for instance) then the client software used for browsing the social network can automatically lower her working level.

### 6.3 Protected Nodes

Privacy levels are assigned to the nodes and their content as follows:
- User *u* defines the privacy level protecting the fact *node(u)*. This privacy level defines the *search policy* [4]. It says which categories of users can reach a search listing of user *u* by performing a global name search. For example, in Figure 3, everybody can reach a search listing of Bob, Alice and Jane, but only people who are at least contacts of Henry's contacts can reach a search listing of Henry.
- User can write messages on their own node or on other users' node. The privacy level protecting each new message is equal to the working level of the user who wrote the message. This working level should dominate the privacy level of the node where the message is written. It should also be dominated by the clearance level of the user writing the message (see section 6.2).

## 7 Mandatory Privacy Policy

Basically, the privacy policy of our model implements the No-Read-up and the No-Write-down principles of the Bell & LaPadula Model. This policy is mandatory. This means it applies to all users and cannot be modified by users. Like in [4], we make a distinction between the search policy, the access policy and the traversal policy. We did not include the communication policy in our model since we would manage it exactly as Facebook currently does.

### 7.1 Search Policy

Users can reach a search listing of node *u* if their clearance level dominates the privacy level protecting the fact $node(u)$. For example all users can reach a search listing of Alice, Bob and Jane but only contacts of Henry's contacts can find him (see Figure 3).

### 7.2 Access Policy

Users can read and write messages on the basis of the Bell & LaPadula principles.
- **No-Read-up:** Users can read a message if their working level dominates the privacy level of the message.
- **No-Write-down:** Users can write messages if their working level is dominated by the privacy level of the message.

Regarding the write operation, we can however make the following comments:
- Write-ups could be disabled in order to avoid low level users spamming high level users.
- If write-ups are enabled, then they should of course be blind. This means that low level users can create messages to be seen by high level users, but cannot update nor delete high level messages since it would require seeing them first.
- Write operation could be restricted to contacts only, or to contacts and contacts of contacts, in order to avoid spamming or integrity attacks from unknown users.

### 7.3 Traversal Policy

The knowledge *v* is a contact of *u* leads to the knowledge *u* is a contact of *v*. This inference problem is known as the problem of the "mutual friend", which exists in the current Facebook system. More precisely, this problem arises when a mutual contact *w* of two contacts (*u* and *v*) is permitted to see one side of the relationship, $contact(u,v)$, and is forbidden to see the other side of the relationship, $contact(v,u)$. From the knowledge $contact(u,v)$, user *w* can learn that *u* and *v* are a contact despite the fact that it is considered as sensitive by user *u* who protected $contact(v,u)$. In order to avoid this problem, we adopt the following traversal policy rule:

**Traversal Policy**: If a user *w* has a privileged relationship with both user *u* and user *v* (i.e. *w* has a relationship with *u* dominating the relationship that *v* has with *u*, and *w* has a relationship with *v* dominating the relationship that *u* has with *v*) then user *w* is permitted to know that user *v* and user *u* are contacts of each other.

As an example, consider Figure 6.
- Alice sees Jane as a Friend, and Jane sees Alice as a Colleague. Alice sees Bob as a Friend, and Jane sees Bob as a Close Friend. Since $Friend \geq Friend \wedge CloseFriend \geq Colleague$, Bob is permitted to see the fact that Alice and Jane are contacts.
- Alice and Bob see each other as Friend. Both Alice and Bob see Jane as a Friend. Since $Friend \geq Friend$, Jane is permitted to see the fact that Alice and Bob are contacts.
- Bob sees Jane as a Friend, and Jane sees Bob as a Close Friend. Bob sees Alice as a Friend but Jane sees Alice as a Colleague. Since $\neg(Colleague \geq CloseFriend)$, Alice is not permitted to see the fact that Bob and Jane are contacts.

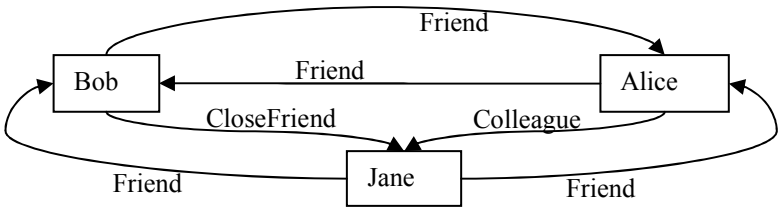


**Fig. 6.** Clearance levels between "mutual friends".

## 8 Tool

Using the Facebook API [11], we have implemented a tool based on our model. The purpose of this tool is to automatically display how the Facebook privacy settings should be set to implement the multilevel privacy policy (No-Read-up and No-Write-down) at the node level. Indeed, in [12], the authors show how to model mandatory policies with roles. Considering the fact that Facebook lists can be seen as roles, our tool applies the principles described in [12] to automatically configure the Facebook settings enforcing the mandatory privacy policy of our model.

Let *u* be a Facebook node,

- User *u* creates 7 Facebook *user lists* corresponding to the 7 privacy levels defined in section 5.1. Note that Facebook user lists cannot be organized into a lattice.
- User *u* distributes her contacts into these Facebook lists as if they were organized into the lattice shown in Figure 2.
- Using our tool, user *u* assigns a virtual privacy level to each data item of her Facebook *profile*.
- The Facebook *wall* is seen as a global data item. User *u* defines the virtual privacy level protecting her wall.

The tool computes then the Facebook privacy settings implementing the No-Read-up and No-Write-down properties of the Bell & LaPadula model. Basically, our tool works as follows:

- **No-Read-up**: Based on Figure 2, it grants the right to see a data item (whether it is a profile item or the wall) to every Facebook user list which dominates the virtual privacy level protecting the data item.
- **No-Write-down**: still based on Figure 2, it grants the right to post messages on the wall to every Facebook user list which dominates the virtual level protecting the wall. It simply assumes that a user with a clearance level dominating the level protecting the wall would systematically lower her working level to the level protecting the wall in order to post a message. For the sake of simplicity, write-ups are disallowed, i.e. users with a clearance level lower than the level protecting the wall cannot post messages on the wall.

Regarding our tool, we can make the following comments:

- The Facebook API does no offer any solution for updating the privacy settings from an application. Therefore, our tool only displays how the Facebook privacy should be set.

– In the current Facebook system, users cannot hide some of their friends from some of their other friends. Therefore, our tool cannot derive the settings which would protect the relationships between users.

## 9 Related Work

In this section, we review several existing access control models for social networks. In [2][7], the authors propose a discretionary security model which adopts a rule-based approach for specifying access to node resources. Like our model, their model takes into account the type of relationships among users. Authorized users are denoted in terms of the type, maximum depth and minimum trust levels of the relationships. For example, by writing the rule $(rid, \{(A, friendOf, 3, 0.8)\})$, Alice expresses the fact that resource *rid* should be available only to her direct and indirect friends with a maximum depth equal to 3 and a minimum trust level equal to 0.8. In [5], the same authors propose an extension to their model for protecting the relationships. If two nodes come into relationship then they negotiate a *distribution rule* specifying the characteristics of the nodes that are authorized to know the existence of the relationship. The relationship is then described in a certificate whose key is protected by the distribution rule like any other resource. While the model described in [2][7][5] provides the users with a highly expressive security policy, we fear that most users would hardly understand the semantics of an access condition consisting of a relationship type, a depth level in a graph, and a trust level. Moreover, as we already underlined it at several occasions in this paper, we do not believe in a model where users, who are not security specialists, need to manage their privacy policy. In [8], the authors propose a multilevel security model for social networks which has some similarities with our model. First, the privacy policy is mandatory i.e. accesses to objects are controlled by strictly enforcing the Bell & LaPadula properties. Second, classification levels of objects are specified by the creators. However, regarding the specification of clearance levels, there is a major difference between their model and our model. In [8], the clearance level of a user is computed as the average trust rating specified for him/her by other users. This computation does not consider at all the type of relationships between users. Therefore, their model seems more suitable to a peer to peer file sharing system where edges between nodes do not represent privileged relationships between users. In [9], the same authors propose a second more sophisticated access control model for social networks. In this model, Alice sets some initial parameters for computing three protection zones per data object: accept zone, attestation zone, and deny zone. Users falling into the accept zone category get unconditional access to Alice's object. Users in the deny zone do not get any access. Requests from users in the attest zone are validated on a per-request basis. Each zone depends on the classification level that Alice has assigned to the data object, on the trust levels that Alice has assigned to some users, on the type of relationships that Alice has with some users, on the hop distance between Alice and the users in the social graph, and on some experiential data resulting from the users' prior actions. Whenever Alice publishes an object, she has to specify a list of attesters. Attesters are some of the attest zone members to whom Alice may grant access to her data object

provided they fulfill some conditions specified by Alice. While this model shows some interesting features and may be a good candidate for a "social" network where companies share online resources, we think it is not appropriate for a traditional social network for ordinary people. Although the authors claim their model is user-friendly, we have difficulty to figure out how users could maintain an attesters' list and an attestation procedure for each object. Finally, let us mention that all the models reviewed in this section do not say much about the write privilege. In particular they seem to ignore the fact that in a traditional social network like Facebook, users can share objects on their contacts' page.

## 10 Conclusions

In this paper we defined a new privacy model for Facebook-style social networks. Whereas the Facebook privacy model is a discretionary model, our model relies on a mandatory policy. Basically, the only security task that users need to do on a regular basis is to categorize each new contact. Except the privacy level defining the search policy, there are no other settings to configure. Moreover, our model fully takes into account the protection of the relationships among users. We formalized our model in first-order logic and implemented it as a Prolog knowledge-base. We also developed a tool allowing us to automatically derive the Facebook settings implementing the multilevel privacy policy. However, with the current Facebook implementation, we cannot derive settings protecting relationships. We believe our model can be a good alternative to the current Facebook model. Of course, it needs some refinements that we could not discuss in this paper due to space limitation. In particular, we did not consider the fact that messages written on Facebook pages mention the name of the user who wrote the message. Therefore, a first refinement could be to state a rule saying that Bob cannot write a message on Alice's page at a level which would disclose the fact that he is Alice's friend. A second refinement could be to consider the special data object referred to as *tag*: a tag is a kind of pointer that can be attached to objects (generally a picture) to reference a particular user. Another refinement could be to consider that users can always delete and update their own messages even in case of write-ups. We are, in fact, currently extending our model to include all these refinements.

## References

1. Katherine Strater and Healther Richter. "Examining Privacy and Disclosure in a Social Networking Community". Proceedings of the 3rd symposium on Usable privacy and security. SOUPS 2007. Pittsburgh, Pennsylvania. 157-158.
2. Barbara Carminati, Elena Ferrari and Andrea Perego. "Enforcing Access Control in Web-based Social Networks". ACM Transactions on Information and System Security (TISSEC). Volume 13 Issue 1, October 2009. Vol 13(1). 1-38.
3. D. Bell and L. LaPadula. Secure Computer Systems: Unified Exposition and Multics Interpretation. Technical Report ESD-TR-75-306, MTR 2997, MITRE, Bedford, Mass. 1975.

4. Fong, Philip, Anwar, Mohd, Zhao, Zhen. A Privacy Preservation Model for Facebook-Style Social Network Systems. Computer Security – ESORICS 2009. LNCS vol 5789. Springer. 303-301.
5. Barbara Carminati, Elena Ferrari and Andrea Perego. Private Relationships in Social Networks. Data and Knowledge Engineering, Vol 37/2, 2001, pp 177-201. Elsevier.
6. A. Gabillon. Web Access Control Strategies. Second edition of Encyclopedia of Cryptography and Security at Springer. Tilborg, Henk C.A. van; Jajodia, Sushil (Eds.). Due June 2011
7. Barbara Carminati, Elena Ferrari and Andrea Perego. Rule-based Access Control for Social Networks. In On the move to Meaningful Internet Systems. OTM 2006 workshops. LNCS, vol 4278. Springer, 1734-1744.
8. Bader Ali, Wifred Villegas and Muthucumaru Maheswaram. "A Trust Based Approach for Protecting User Data in Social Networks. In 2007 Conference of the Center for Advanced Studies on Collaborative Research, CASCON 2007. ACM Press, 288-293.
9. Bader Ali, Wifred Villegas and Muthucumaru Maheswaram. "An Access Control Scheme for Protecting Personal Data". Sixth Annual Conference on Privacy, Security and Trust, PST 2008, October 1-3, 2008, Fredericton, New Brunswick, Canada. 24-35.
10. SWI-Prolog. http://www.swi-prolog.org/
11. Facebook Developpers. http://developers.facebook.com
12. Sylvia Osborn, Ravi Sandhu and Qamar Munawer. Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. In ACM Transactions on Information and System Security, Vol. 3, No. 2, May 2000, Pages 85–106.