IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 2 (Jul. - Aug. 2013), PP 01-07
www.iosrjournals.org

Privacy Management of Multi User Environment in Online Social Networks (OSNs)

P. Amrutha[1], R. Sathiyaraj[2]
M.Tech Scholar , Dept.of CSE, Madanapalle Institute of Technology and Sciences, Madanapalle, JNTUA[1]
Assistant professor, Dept.of CSE, Madanapalle Institute of Technology and Sciences, Madanapalle, JNTUA[2].

Abstract: *Online Social Networks (OSNs) are inherently designed to enable people to share personal and public information and make social connections with others. These OSNs provides digital social interactions and social as well as personal information sharing, but in sharing a number of security and privacy problems raised. While OSNs allow users to restrict access to shared data, they currently do not provide any mechanism to totally enforce privacy issue solver associated with multiple users. To this end, we propose an approach to enable the protection of shared data associated with multiple users in OSNs. We formulate an access control model to capture the essence of multiparty authorization requirements, along with a multiparty policy specification scheme and a policy enforcement mechanism. Besides we also implement a proof-of-concept prototype which is called as MController (multi controller) having contributor, stakeholder and disseminator controllers along with owner controller.*

Index Terms *--- social network, multi party access control, MController, decision voting*

I. Introduction

Many people interested to share personal and public information and make social connections with friends, family, colleagues, coworkers and even with strangers through Online Social Networks(OSN) such like Facebook, Twitter, Google+ and etc,. OSN provide some space to each user for basic profile and sharing photos and videos with others. In photo sharing unfortunately some privacy and security problems are raised. Presently there is no mechanism to totally avoid these privacy issues. The main problem is collaborative authorization management, means if user tags the photo to his friend only. But the updates of photo are presented in both user as well as friends profiles. Then friend of friends or others may share that photo. So here the user expected privacy was spoiled. The existing protection for photos is binary condition either put or delete in profile space. If the photo was deleted after tagging, the content may loss in space, else the privacy was spoiled.

1.1. OSNs Privacy

In OSNs privacy restrictions form a spectrum between public and private data. On the public end, users can allow every particular OSN member to view their personal content. On the private end, users can restrict access to a specific set of trusted users. Despite the spectrum of available privacy settings, users have no control over information appearing outside their immediate profile page, when a user comment on friend's image, user and friend both cannot restrict the comment from other viewers. Similarly, if a user posts a photo and indicates the name of a friend in the photo, the friend cannot specify which users can view the photo. For both of these cases, Facebook currently lacks a mechanism to satisfy privacy constraints when multiuser is involved, So that the user's privacy may be violated. Privacy conflicts publicly expose personal information, slowly decreasing a user's privacy.

The user would have more control over his photos where a set of malicious users may want to make a shared photo available to a wider audience. If the malicious users can access the photo from original user then they tag photo with fake identities to others. Those may further share with other users. This continuous process, by this the original photo may change totally and shared with number of persons. At that time the privacy of photo which was expected by original user may collude totally. To prevent such an attack, three conditions need to be satisfied:

- No Fake Identity in OSNs.
- All Tagged Users are Real Users for the Photo.
- All Controllers are Honest to specify their Privacy policies for the photo.

II. MController

OSN is mainly relationship network including set of users as well as their data. So that OSN represented with directed labeled graph where each node represents user and edge denotes relationship between two users. The edge direction denotes the relationship from initial to terminal node. The profile space of the user managed himself with his privacy data and content. For that privacy data to maintain security several schemes are introduced. But no scheme gives totally security, mainly all those schemes have only one controller that is owner. By this single controller security and privacy issues may be raised on data which was personal to the owner.

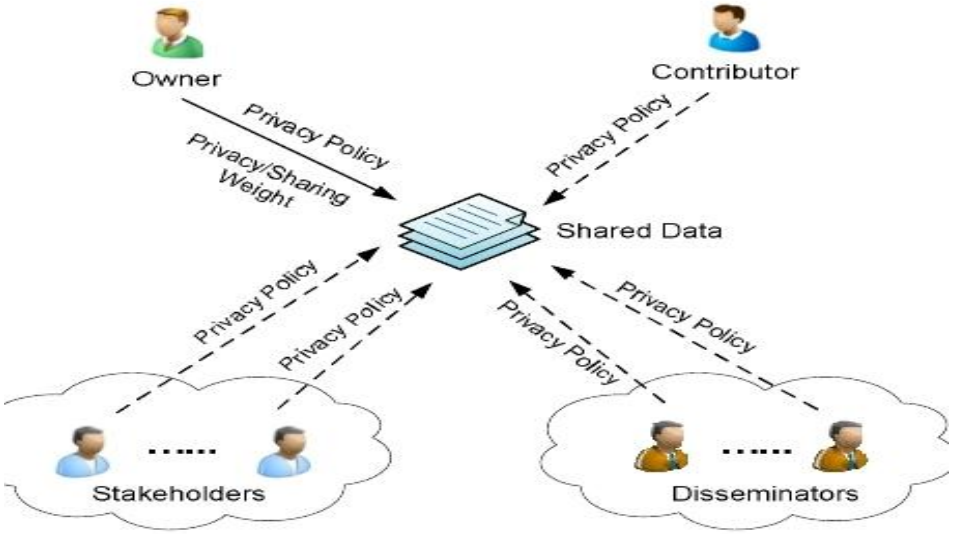


Figure.1. MController Architecture

So that rather than the owner controlling additional controllers are need for the flexible privacy mechanisms in OSN. The additional controllers are contributor, stakeholder and disseminator which provide their own privacy policies on shared data by giving the permission either permit or deny to unauthorized user on shared data. Figure 1 illustrates different controllers providing their privacy policies on shared data. We define multi controllers as follows:

- **Owner (O):** In the social network the user u is called the owner of the data item d, if d presents in the space m of user u. The user u is also called as contributor of d, when that user share data item d. The owner share data in three types, they are profile sharing, content sharing and relationship sharing. It enables the owner to discover potential malicious activities in collaborative control.
- **Contributor (C):** In the social network the user u is called the contributor of the data item d, if d published by user u in someone else's space. The contributor tags content to other's space and the content may also have multiple stakeholders (e.g., tagged users). The memory space for the user will be allotted according to user request for content sharing.
- **Stakeholder (S):** In the social network the user u is called a stakeholder of the data item d, if user u is tagged user T for d. A shared content has multiple stakeholders.
- **Disseminator (D):** In the social network, let d be a data item shared by a user u from someone else's space to his/her space. The user u is called a disseminator of d. the real content sharing starts with the owner, then disseminator views the content and shares with others. This disseminated content may be re-disseminated again and again by others.

III. Multi Party Access Control (MPAC) Model

3.1. MPAC Specification

It is very essential for MPAC policies to regulate access and representing authorization requirements from multiple associated users to enable a collaborative authorization management of data sharing in OSNs.

- **Accessor Specification:** Accessor is the set of users who granted to access the shared data. Accessor can be represented with a set of user names, relationship names and group names in OSNs.

The accessor specification is defined as a set, accessors = {a1, a2. . . , an}, where each element is a tuple < ac,at >. where ac ∈ U ∪ RT ∪ G be a user u ∈ U, a relationship type rt ∈ RT, or a group g ∈ G. at ∈ {UN,RN,GN} be

the type of the accessor specification, where UN,RN,GN represents user name, relationship name, and group name.

- **Data Specification:** The data specification represented in three ways; profile, relationship and content sharing. For effective privacy the different controllers provide sensitivity levels on data.
  Let dt ∈ D be a data item, sl be a sensitivity level (range 0.00 to 1.00) for data item dt. The data specification is defined as a tuple < dt, sl >.

3.2. MPAC Policy

To summarize the above-mentioned specification elements, we introduce the definition of a multiparty access control policy as follows:
The multi party access control policy is a 5 - tuple
P = < controller, Ctype, accessor, data, effect >
where

- Controller is a user who can regulate the access of data.
- Ctype is the type of the controller.
- Accessor is the set of users who granted to access the shared data.
- Data is represents a data specification.
- Effect ∈ {permit, deny} is the authorization effect of the policy. Suppose a controller can leverage five sensitivity levels: 0.00 (none), 0.25 (low), 0.50 (medium), 0.75 (high), and 1.00 (highest) for the shared data.

3.3. MPAC Evaluation

Multi party access control is evaluated in two steps. In step-1, the individual decision are collected from different controllers, and in step-2, individual decision are aggregated and makes final decision for the access request.

Figure 2 illustrates that how MPAC evaluated in step by step. Initially an access request goes to under policy evaluation, which is done under four controllers. The four controllers provide their own privacy policies in the form of decision either permit or deny in step-1 process. After giving decisions by individual controllers, they are aggregated and make final decision by using decision voting schemes in step-2 process. The final decision making decides whether the access request is allowed or refused.

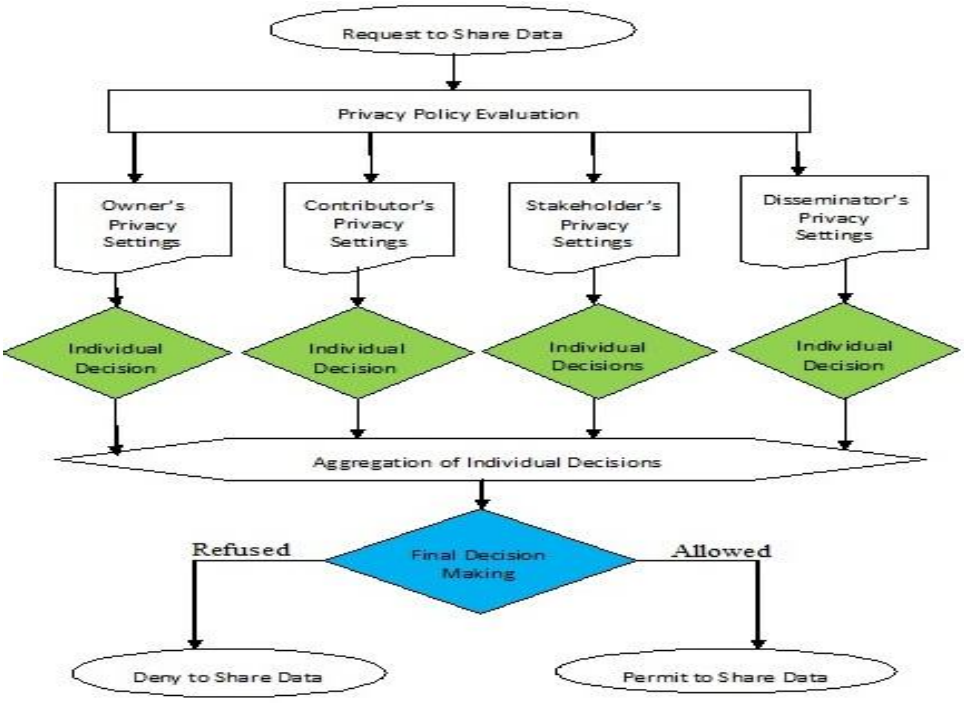


Figure.2. MPAC Evaluation

From the process of evaluation in MPAC policies, the controllers give different decision for an access request. There may be a chance of occurring conflicts. So that a mechanism is needed to resolute the conflicts

for taking an unambiguous decision for each access request. For the better privacy, a strong resolution for conflict may need. So it is better to consider tradeoff between privacy and utility in resolution of conflict. For this conflict issue, we introduce decision voting schemes resolving the MPAC conflicts which is simple and flexible.

IV. Final Decision Making Schemes

4.1. Decision Voting Mechanism

Decision making mainly depends on majority. For such decision making, we introduce a voting scheme for conflict resolution. In voting mechanism each controller's individual decision effects the final decision. Mainly this voting scheme is described in two voting mechanisms; they are decision voting and sensitivity voting.

4.1.1. Decision Voting: the policy evaluation derives the decision voting value (DV) either permit or deny as follows,

Where Evaluation(p) represents the policy p decision:

$$DV = j \begin{cases} 0 & \text{if Evaluation(p) = Deny} \\ 1 & \text{if Evaluation(p) = Permit} \end{cases} \quad (1)$$

Assume that all controllers are equally important, an aggregated decision value (DVag) (range 0.00 to 1.00) from multiple controllers including the owner (DVow), the contributor (DVcb) and stakeholders (DVst), is computed with following equation:

$$DVag=(DVow+DVcb+\sum_{i\in SS} DV^{i}_{st})\times 1/m \quad (2)$$

Where SS is the set of stakeholders for shared data item, and m is total number of controllers for shared data item.

For the shared data item each controller may have (i) a different trust level over the data owner and (ii) a different reputation value in terms of collaborative control. So we need to introduce weights for decision voting scheme. Weights for different controllers can be calculated by aggregating trust levels and reputation values. The weight of controller x is "weightx / sum of weights". Suppose ωow, ωcb and ωi sh are weights for owner, contributor and stakeholder controllers, respectively, and n is the number of stakeholders of the shared data item. A weighted decision voting scheme is as follows:

$$DVag=(\omega ow\times DVow+\omega cb\times DVcb+\sum_{i=1ton}(\omega i_{st}\times DV^{i}_{st}))\times 1/(\omega ow+\omega cb+\sum_{i=1ton}\omega ist) \quad (3)$$

4.1.2. Sensitivity Voting: Each controller assigns a sensitivity level (SL) to the shared data item to reflect her/his privacy concern. A sensitivity score (SC) (range 0.00 to 1.00) for the data item can be calculated based on following equation:

$$SC=(SLow+SLcb+\sum_{i\in SS} SL^{i}_{st})\times 1/m \quad (4)$$

4.2. Threshold-Based Conflict Resolution

A basic idea of our approach for threshold-based conflict resolution is that the sensitivity score (SC) can be utilized as a threshold for decision making. Obviously, if SC increased, then the chance of final decision to deny is increased, so that the utility of OSN services cannot be affected. The threshold-based conflict resolution calculates final decision as follows:

$$Decision = j \begin{cases} Permit & \text{if } DVag > SC \\ Deny & \text{if } DVag \leq SC \end{cases} \quad (5)$$

It is worth noticing that our conflict resolution approach has an adaptive feature which reflects the changes of policies and sensitivity levels. If any controller changes his privacy policy or sensitivity level on the shared data item, then the aggregated decision value (DVag) and the sensitivity score (SC) will be recomputed and accordingly the final decision may be changed.

4.3. Strategy-Based Conflict Resolution

If we treat all controllers equally important, then above threshold-based conflict resolution provides a simple mechanism for making final decision. But in practical, different controllers may have different priorities making final decision. Especially the owner has highest priority in the control of shred data item. So that we provide strategy-based conflict resolution mechanism to satisfy owner authorization requirements of shared data.

Here the sensitivity score (SC) considered as guideline in selecting appropriate strategy for conflict resolution of shared data item. We introduce following strategies for the purpose of resolving multiparty privacy conflicts in OSNs.

- **Owner−overrides:** In final decision making, the highest priority goes to owner's decision. This strategy is totally owner controlling mechanism in data sharing. Based on the weighted decision voting scheme, we set ωow = 1, ωcb = 0 and ωst = 0,1 and the final decision can be made as follows:

$$\text{Decision} = j \begin{cases} \text{Permit} & \text{if DVag} = 1 \\ \text{Deny} & \text{if DVag} = 0 \end{cases} \quad (6)$$

- **Full−consensus−permit**: The final decision is deny, if any controller deny the access. This strategy can achieve the naive conflict resolution. The final decision can be derived as:

$$\text{Decision} = j \begin{cases} \text{Permit} & \text{if DVag} = 1 \\ \text{Deny} & \text{otherwise} \end{cases} \quad (7)$$

- **Majority−permit:** This strategy permits (denies, resp.) a request if the number of controllers to permit (deny, resp.) the request is greater than the number of controllers to deny (permit, resp.) the request. The final decision can be made as:

$$\text{Decision} = j \begin{cases} \text{Permit} & \text{if DVag} \geq ½ \\ \text{Deny} & \text{if DVag} < ½ \end{cases} \quad (8)$$

V. Logical Representation of Multiparty Access Control

We introduce an ASP program for multiparty authorization specification.

5.1. Logical Definition of Controllers and Relationships

The basic components and relations in our MPAC model can be directly defined with corresponding predicates in ASP. We have defined UDct as a set of user-to-data relations with controller type ct ∈ CT. Then, the logical definition of multiple controllers is as follows:

• The owner controller of a data item can be represented as:
OW(controller,data) ← UD_{OW}(controller, data) ∧ (controller) ∧ D(data).
• The contributor controller of a data item can be represented as:
CB(controller, data) ← UD_{CB}(controller, data)∧ U(controller) ∧ D(data).
• The stakeholder controller of a data item can be represented as:
ST(controller, data) ← UD_{ST} (controller, data)∧ U(controller) ∧ D(data).
• The disseminator controller of a data item can be represented as:
DS(controller, data) ← UD_{DS}(controller, data)∧ U(controller) ∧ D(data).
Our MPAC model supports transitive relationships. Then, friends-of-friends can be represented as a transitive closure of friend relation with ASP rule as follows:
friendsOFfriends(U1, U2) ← friendOf(U1, U2).
friendsOFfriends(U1, U3) ← friendsOFfriends(U1, U2), friendsOFfriends(U2, U3).

5.2. Logical Representation of Decision Voting Schemes

decision voting(C) = 1 ← decision(C, permit).
decision voting(C) = 0 ← decision(C, deny).
aggregation weight(K) ← K = sum{weight(C) : controller(C)}.
aggregation decision(N) ← N = sum{decision voting(C) × weight(C) : controller(C)}.
aggregation sensitivity(M) ← M = sum{sensitivity voting(C) × weight(C) : controller(C)}.

5.3. Logical Representation of Threshold-Based Conflict Resolution

decision(controllers, permit) ←N >M ∧ aggregation decision(N) ∧ aggregation sensitivity(M).
decision(controllers, deny) ← not decision(controllers, permit).

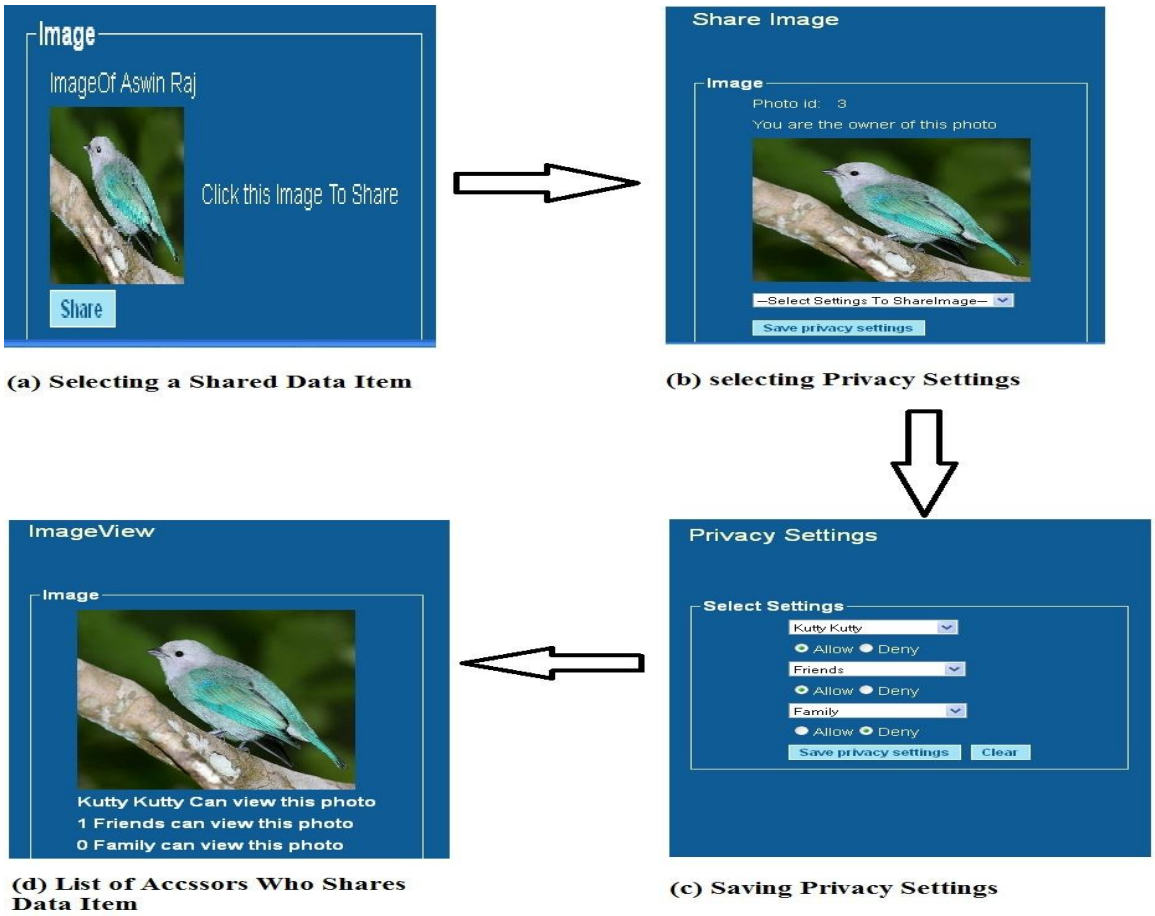
5.4. Logical Representation of Strategy-Based Conflict Resolution

• The conflict resolution strategy for Owner−overrides is represented as:
weight(controllers) = 1 ← OW(controller, data).
weight(controllers) = 0 ← CB(controller, data).
weight(controllers) = 0 ← ST(controller, data).
decision(controllers, permit) ← N/K == 1∧ aggregation weight(K) ∧aggregation decision(N).
decision(controllers, deny) ← not decision(controllers, permit).
• The conflict resolution strategy for Full−consensus−permit is represented as:
decision(controllers, permit) ← N/K == 1∧ aggregation weight(K) ∧aggregation decision(N).
decision(controllers, deny) ← not decision(controllers, permit).

• The conflict resolution strategy for Majority−permit is represented as:
decision(controllers, permit) ← N/K > 1/2 ∧ aggregation weight(K) ∧ aggregation decision(N).
decision(controllers, deny) ← not decision(controllers, permit).
• The conflict resolution strategy for Deny−overrides for dissemination control is represented as:
decision(deny) ← decision(controllers, deny).
decision(deny) ← decision(disseminator, deny).
decision(permit) ← not decision(deny).

VI. Implementation

MController is third-party application development for Facebook. This is hosted in an Apache Tomcat application server supporting PHP and MySQL database. MController application is based on the iFrame external application approach. Using the Javascript and PHP SDK, it accesses users' Facebook data through the Graph API and Facebook Query Language. Once user install MController in his Facebook space and accepts the terms and conditions, then MController access the content and basic information of user. Mainly, it retrieves the list of all photos owned by user as well as tagged photos and uploaded. Now user access MController privacy settings on shared images and protect from other viewers.



(a) Selecting a Shared Data Item

(b) selecting Privacy Settings

(d) List of Accssors Who Shares Data Item

(c) Saving Privacy Settings

Figure.3. Snapshot of MController

A snapshot of main interface of MController is shown in Figure 3 illustrates that how MController runs in each step and execution. Initially the user selects the image which he needs share and click on share button as showing in figure 3.a. the figure 3.b shows share image and privacy setting option. This privacy setting option is the main aim of MController system. If the user selects the privacy settings option, settings page appeared as like figure 3.c. the figure 3.c shows the options of individual persons as well as groups. Here the user selects access or deny option for different groups like family, friends and coworkers. After settings completed, the user click on save button or else click on cancel button to reset the settings. Once the settings are saved by user, under the shared image, the list of visitors can appeared according to the user privacy settings as shown in figure 3.d. The visitors list informed that who can only see and share the user's image.

VII. Related Work

Access control for OSNs is still relatively a new research area for privacy issues. Presently several access control models for OSNs have been introduced. Fong et al. proposed an access control model that formalizes and generalizes the access control mechanism implemented in Facebook, which admitting arbitrary policy vocabularies that are based on theoretical graph properties. Fong recently formulated this paradigm called a Relationship- Based Access Control (ReBAC) model that bases authorization decisions on the relationships between the resource owner and the resource accessor in an OSN. Carminati et al. recently introduced collaborative security policies, a new class of security policies, that basically enhance topology-based access control with respect to a set of collaborative users.

VIII. Conclusion

In this paper, we found the need of privacy for OSN and solution of collaborative authorization management of the shared data. We introduced MController technique to provide their own privacy preferences on a shared data by the different controllers. Additionally MPAC model evaluated providing decision voting schemes and the privacy evaluation. In the future work, we are planning to investigate advanced MController technique to provide privacy settings for the group of photos at a time, because users may be involved to put privacy setting for the number of photos at a time. By this MPAC model it is time consuming process. So that we would study advanced MController for shared data to automatic configure the privacy.

References

[1] Hongxin Hu, Gail-Joon Ahn, Senior Member, IEEE, and Jan Jorgensen "**Multiparty Access Control for Online Social Networks: Model and Mechanisms"** IEEE transactions,2012.

[2] Besmer and H. Richter Lipford. **"Moving Beyond Untagging: Photoprivacy in A Tagged World"**. pages 1563–1572. ACM, 2010.

[3] L. Bilge, T. Strufe, D. Balzarotti and E. Kirda. **"All Your Contacts are Belong to Us: Automated Identity Theft Attacks On Social Networks"**. pages 551–560. ACM, 2009.

[4] Carminati and E. Ferrari. **"Collaborative Access Control in Online Social Networks"**. pages 231–240. IEEE, 2011.

[5] Carminati, E. Ferrari, and A. Perego. **"Rule-Based Access Control for Social Networks"**. pages 1734–1744. Springer, 2006.

[6] B. Carminati, E. Ferrari, and A. Perego. **"Enforcing access control in web-based social networks."** ACM Transactions on Information and System Security (TISSEC), 13(1):1–38, 2009.

[7] Carrie. **"Access Control Requirements for Web 2.0 Security and Privacy."** In Proc. of Workshop on Web 2.0 Security & Privacy (W2SP). Citeseer, 2007.

[8] J. Choi, W. De Neve, K. Plataniotis, and Y. Ro. **"Collaborative face recognition for improved face annotation in personal photo collections shared on online social networks."** Multimedia, IEEE Transactions on, 13(1):14–28, 2011.

[9] P. Fong. **"Preventing sybil attacks by privilege attenuation: A design principle for social network systems."** In Security and Privacy (SP), 2011 IEEE Symposium on, pages 263–278. IEEE, 2011.

[10] P. Fong.**"Relationship-based access control: Protection model and policy language".** In Proceedings of the first ACM conference on Data and application security and privacy, pages 191–202. ACM, 2011.

[11] P. Fong, M. Anwar, and Z. Zhao. "**A privacy preservation model for facebook-style social network systems".** In Proceedings of the 14th European conference on Research in computer security, pages 303–320. Springer-Verlag, 2009.

[12] J. Golbeck. "**Computing and applying trust in web-based social networks"**. Ph.D. thesis, University of Maryland at College Park College Park, MD, USA. 2005.

[13] M. Harrison, W. Ruzzo, and J. Ullman. "**Protection in operating systems"**. Communications of the ACM, 19(8):461–471, 1976.

[14] H. Hu and G. Ahn. "**Enabling verification and conformance testing for access control model**". In Proceedings of the 13th ACM symposium on Access control models and technologies, pages 195–204. ACM, 2008.

[15] H. Hu and G. Ahn. "**Multiparty authorization framework for data sharing in online social networks**". In Proceedings of the 25th annual IFIP WG 11.3 conference on Data and applications security and privacy, pages 29–43. Springer-Verlag, 2011.

[16] H. Hu, G. Ahn, and K. Kulkarni. "**Anomaly discovery and resolution in web access control policies**". In Proceedings of the 16th ACM symposium on Access control models and technologies, pages 165– 174. ACM, 2011.

[17] H. Hu, G.-J. Ahn, and J. Jorgensen. "**Enabling Collaborative Data Sharing in Google+**". Technical Report ASU-SCIDSE-12-1, April 2012.

[18] H. Hu, G.-J. Ahn, and J. Jorgensen. "**Detecting and resolving privacy conflicts for collaborative data sharing in online social networks**". In Proceedings of the 27th Annual Computer Security Applications Conference, ACSAC '11, pages 103–112. ACM, 2011.

[19] H. Hu, G.-J. Ahn, and K. Kulkarni. "**Detecting and resolving firewall policy anomalies. IEEE Transactions on Dependable and Secure Computing**", 9:318–331, 2012.

[20] L. Jin, H. Takabi, and J. Joshi. "**Towards active detection of identity clone attacks on online social networks**". In Proceedings of the first ACM conference on Data and application security and privacy, pages 27–38. ACM, 2011.

[21] L. Lam and S. Suen. "**Application of majority voting to pattern recognition: an analysis of its behavior and performance**". Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 27(5):553–568, 2002.