JKAU: Comp. IT., **Vol. 2**, pp: 3 – 25 (2013 A.D./ 1435 A.H.)

DOI: 10.4197 / Comp. 2 -1

Protecting User Privacy from Social Networking Applications

Ebtesam A. Alomari, Maram S. Alhafizy, Omar A. Batarfi

Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, P.O.Box. 80221, Jeddah 21589, Kingdom of Saudi Arabia
obatarfi@kau.edu.sa

Abstract. Day by day, users of social network are increasing. They use different applications offered by social networking and publishing more personal information. However, users try to protect their privacy by keeping their information as private or share them only with family members or friends group. Further, many of them are unaware of the degree to which their privacy depends on third party applications and their developers, which left their private information vulnerable to accidental or malicious leaks by these applications. In this work, we focus on users' privacy concerns toward third-party application. We provide methodology to prevent these applications from sharing users' data with friends inside a social network without user permission, by the proposed system to allow the users through the installation process to give different level of permission to the data that is requested by friends' applications. Finally, the proposed system is evaluated for feasibility with theoretical analysis as well as simulation experiments by developing a client server application by Java to simulate a social network environment. Based on testing results, the proposed system assists users to have more control on their data and can achieve the level of privacy that they want.

Keywords: social network, third party application, privacy, security, and friends' applications.

1. Introduction

Social networking services are dominant on the web today, like Facebook, and LinkedIn. These services cater for a broad range of users with different ages, and backgrounds. They allow the users who have limited technical skills to publish personal information and communicate

in an easy manner. People have different motivation to join social networks. They may create a profile for either professional or personal purposes, use the different applications offered by social network and share information with selected contacts or the public [1]. More and more user data are being published which may contain personal information that users may keep as private or share only with family members or friends group. Therefore, preserving user privacy in social network has become an important concern [2].

Social networking applications that facilitate online social interaction and sharing information among a large number of users are increasingly becoming a major type of online applications. These complex environments bring new security and privacy challenges [3]. Users constantly provide information to these applications, which may include privacy-sensitive information.

The main goal of the social network is to encourage the users to expand their social connectivity through interactions and content sharing with each other. Some social networks such as Facebook and MySpace provide some privacy control settings to their users, but some settings like the access and privacy control features are still limited and not flexible and robust, and not appropriate with a huge user base and high volume of privacy-sensitive content [3].

Third-party applications interact with a social network without being part of that social network. Social networking gives these third-party applications access to user data to provide some attractive services; however this may lead to serious privacy risks by exposing user data to these applications. To prevent privacy violations, social networking sites provide users with access control settings to place restrictions on who may view their personal information [4].

Disclosure of information about a user to other members that are not explicitly trusted by the user, without the permission of the user, has to be prevented. Privacy is the possibility to hide any information about any user, unless explicitly disclosed by the users themselves [1]. There are many researches on social networks for producing effective means of ensuring user privacy; most of these researches focused on techniques to prevent identity disclosure and neglected attribute disclosure attacks [5].

This paper focuses on two problems related to user data privacy that may be violated by third party applications in social networks, which are considered as non-trusted side, that may access to user information and share them without any restriction to protect user privacy. The major purpose of this research is to provide sufficient techniques which can secure data from others, protect to access data from other applications, and finally provide users more control over data to not be disclosed to the third-party applications.

2. Literature Review

Heng *et al*. [6] focused on users' privacy concerns toward third-party application on Facebook. They proposed two designs of privacy authorization dialogues, to control the data accessed and used by applications before adding them to the user's Facebook profile. They provided these solutions to address the defect in the current application installation dialogue in Facebook, which allow asking access request to group of different types of information together, such as asking access to all profile information while user may have different preferences for disclosing these different types of data. Therefore, they proposed two-design principles. The first principle indicated that privacy authentication dialogue should provide options for a user to give the permission for each data asked by the application. The second design was alerting user when applications ask for the sensitive private information by providing alert signals. However, there are two limitations in these solutions as we present in Fig.1. Their second proposed design for alert signal is not sufficient. Therefore, users may give permission to the application to share with others their data that they decided it before in privacy setting as private. This means that users may violate their privacy by themselves. In addition, as shown in Fig.1 application may get private information as necessary data, so user may not get attention to that and agree to complete installation process. However, we consider in this paper these two limitations and propose solution to restrict violating users' privacy and prevent disclosing their sensitive private data without permission.

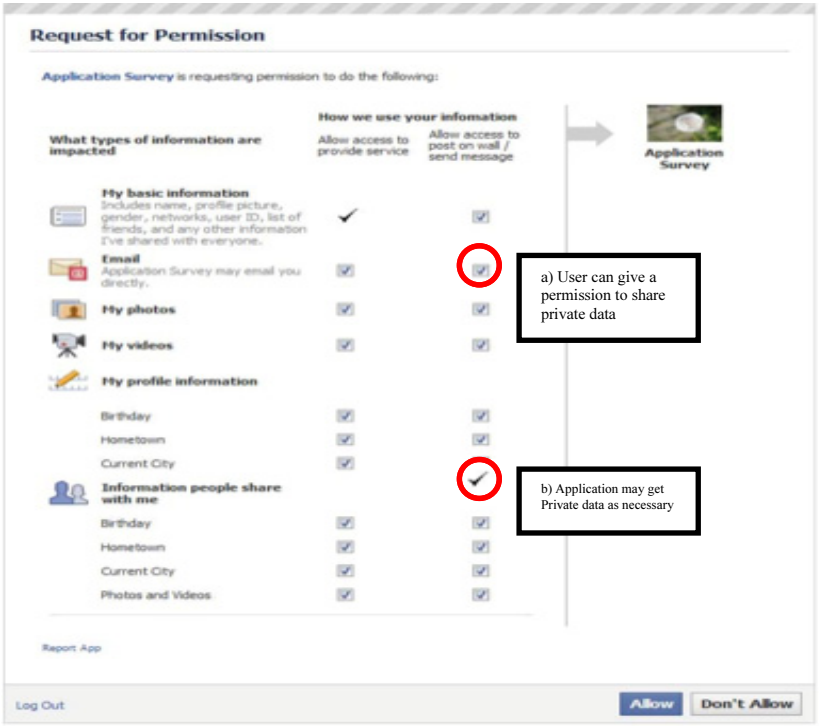


Fig. 1. Limitations in proposed solution by Heng *et al.*[6].

Moreover, Shehab *et al.* [7] illustrated an access control framework to control user data attributes shared with the third-party applications, based on enabling the users to determine the degree of specificity for the shared attributes. They supposed that services of applications are divided into four levels, base on the amount of services that they provide them for the users. These levels are minimum service, intermediate I, intermediate II, and maximum services. Further, they proposed that application informs the user the minimum requirement for each level of services, and then the social network asks users to select the level in which they prefer the application to function. However, in their solution users are not known whether this application need to get access to these data only or use them for share with others. Therefore in our solution, we consider this limitation and suppose levels of permission for each data allowed to the third-parity application, which means that user can decide which data he allows the application to get and for either access only or access and share with allowed friends.

Furthermore, Viswanath *et al*. [8] discussed the problem of sharing personal information with user friends while application can receives data from user friends and forwards them to other friends without their knowledge or permission. They called this attack Kevin Bacon attack and they indicated that there is a need to restrict information flow to protect against this problem. Further, they suggested executing the user principal in two sandboxes. The first one called read-only sandbox, which can read information shared from friends, but cannot write to the user database or share to other friends. The second sandbox can write to the user database and share with friends, but cannot read data shared from friends, which means that the data come from a friend is separated to prevent the Kevin Bacon attack. However, some applications need access to friends data to provide services. In this case our solution indicates that user will be able to decide which data accessed by friends application. Therefore, applications are not able to share user sensitive private data to others without user permission.

3. Problems

Third-party applications in social network are not considered as trustworthy, where they may access to users information and share them without any restriction to protect user privacy. This section discusses two problems related to user data privacy.

3.1 Sharing users data with their friends without permission

A large number of third-party applications needs sharing data with friend as presented by Viswanath *et al.* [8]. They analyzed 50 Facebook applications, including the top 25 applications and the rest were selected randomly from the top 1000 applications. Then, they examined each application whether it shares the data of the user with others or not. Therefore, they found the majority of applications about 52% require sharing data with friends only. This is considered the best behavior in terms of privacy for data sharing. However, this opens the possibility for a new kind of attack when application silently gets information about user and forwards it to a set of friends without user permission.

In social network today, when user wants to install the application, the authentication dialogue will appear to notify users about data that this application want to gets, as shown in Fig. 2.

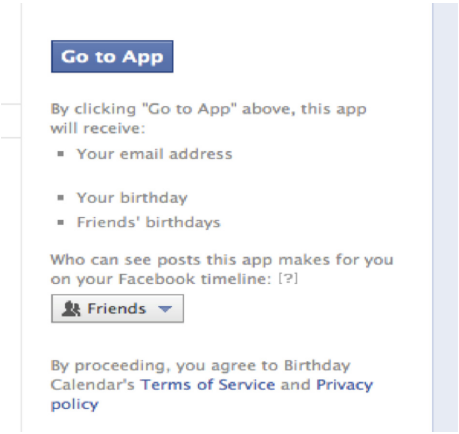


Fig. 2. Current authorization windows in Facebook.

However, during this process users do not have any control to prevent the applications from access to specific information or restrict them from sharing with others in social network. Moreover, the proposed solution by Heng *et al.* [6]. did not prevent application from access to user sensitive private data as presented in section 2.

3.2 Friends' applications get access to user private data

The current social network, such as Facebook, gives user the ability to determine data that are accessed by friends' application as shown in Fig. 3. However, there are two limitations to this specification method: 1) It is applied to all friends' application. 2) It may allow friends' application to access user data not intended for them to perform their function. For example the application may need birthday and photos while user, as shown in Fig. 3, allows them to access by many other data.

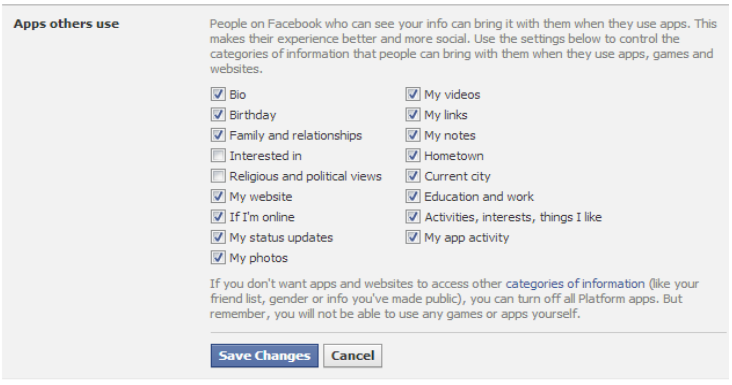


Fig. 3. The policies that apply in Facebook for friends' application.

4. Proposed Solution

Our contribution in this paper is proposing a methodology to counteract privacy loss due to access and share user's data by third-party applications in social networking. The solutions will take into consideration that the functionality of the applications is very important for users and the application developers need to get access to the data to provide services to them. Therefore, we suggest giving the decision for the user to decide the level of permission allowed to these applications. We propose a more efficient control model that allows users to control their data and to protect their privacy with the help of social network, as a trusted party.

Further, in the architecture of our solution, as shown in Fig. 4, the user can control the access rights for applications to his data by giving the permission is stored in social network. Therefore, through access or share request from application to social network, the social network should check the policy that was assigned by the user, stored in policy database, to know the level of permission allowed to this application. So, accessing or sharing user data will not be complete successfully if the application does not have the permission.

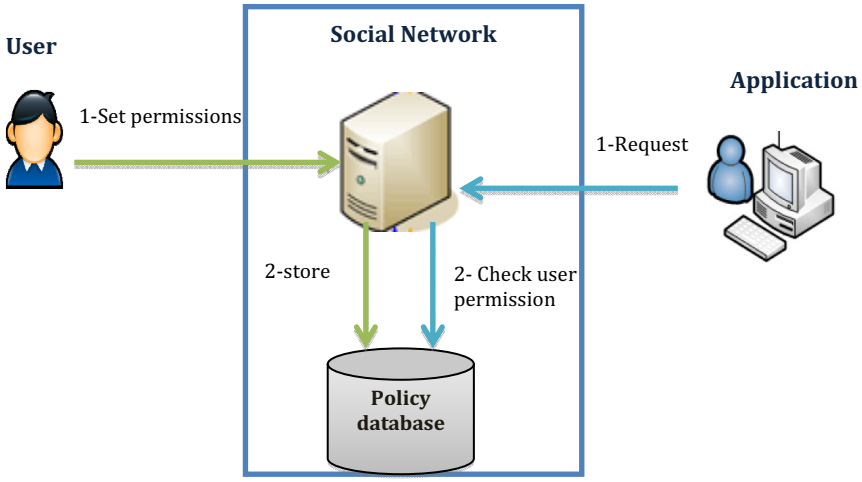


Fig. 4. System architecture.

4.1 Proposed Solution to Problem 1:

If a user does not want friends to know his private data, such as birthday and email, an application that has a permission to access these data should not be able to share them without permission. Therefore, we suggest that when a user wants to install an application, the authentication dialog containing the default setting asked by the application, will appear. It has a drop lists with three options:

- Allow access only: means application is allowed to access only to the data but it cannot share it inside the social network with the user friends.
- Allow access and share: means the application has the permission to access and share the data inside the social network with the user friends.
- Do not allow: means the application is prevented from access or sharing this data.

Accordingly the user could be able to change the permission assigned to data and chooses from the list as shown in Fig. 5. However, the user cannot change the permission "Required Access and Share" that assigned by the application for the required data. Furthermore, to enhance users awareness and protect them from giving the application by mistake, the permission to share their private data with the public in social network, we suggest an improvement in the solution proposed by Heng *et al*. [6] to handle their limitations as the following:

- *Attribute that considered private should have red "i" mark beside it, as shown in Fig. 5, to inform user that it is private and giving permission for application to share it with others will conflict with the previous privacy setting. So, user in this case should change the permission from 'share and access' to 'access only' or select 'don't allow' to prevent application from getting this private data. However, if user does not change the permission, another window will appear before installation completes to alert user, as shown in Fig. 6.*

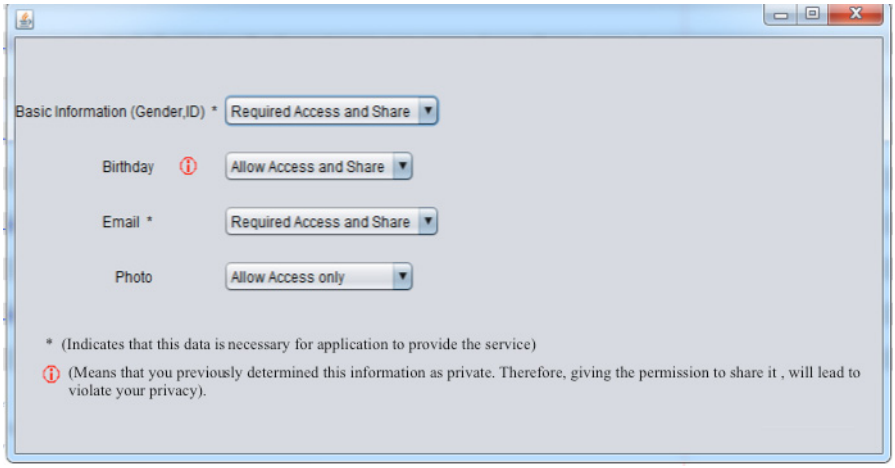


Fig. 5. Control access and sharing permission during installation.

Fig. 6. Alert users to change assigned permission.

- *When application needs a private attribute as a necessary to provide their service, an alert window should appear as shown in Fig. 7, to the user before the installation process complete. To inform user that this application will be able to violate the privacy by getting permission to share private data.*

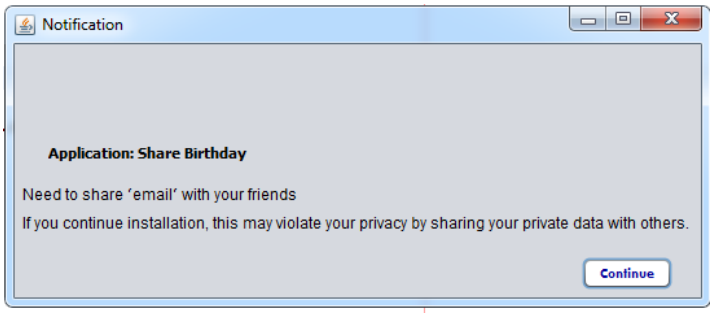


Fig. 7. Alert users during installation.

Therefore, as shown in Fig. 8, the sharing request by application will not perform if it does not have permission. As a result, the alert massages will enhance the users awareness toward their privacy.

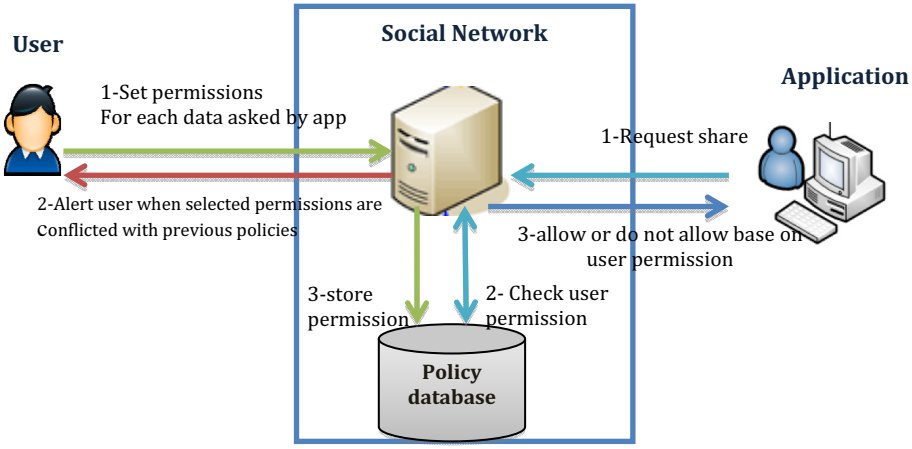


Fig. 8. Architecture for proposed solution-Part. 1.

As shown in Fig. 8 above, our proposed process from the user side is as follows:

1. User sets the permission for each application through the installation process and determines the level of access for each data.
2. Before the installation process complete, an alert message will be sent to user if there is a conflict with previous policies determined by users.
3. The determined permission by users for each data asked by each application will be stored in policy database of the social network.

Further, our proposed process when application intends to share user data with friends is as follows:

1. Application will send the request to social network to share user data with friends.
2. Social network will check the database to know whether user gives permission for sharing these data or not.
3. The social network will inform the application if it does not have a permission to share the data; otherwise the sharing request will perform successfully.

4.2 Proposed Solution to Problem 2:

For this problem, we propose an algorithm to add restrictions for accessing attributes by friends' applications and to address the limitation from specifying the general attributes for all friends' applications, as discussed in section 3. Therefore, we suggest the following:

1) The social network should give the user the ability to determine the level of accessing to the data for friends' applications, either level 1 or level 2.

- Level 1: means friends' applications can access only the attributes necessary to provide the service, and only if it matches the general attributes determined by the user previously.
- Level 2: means friends' applications can access any attributes that match the general attributes determined by the user previously.

2) The social network should give the user the ability to determine either to apply these levels of accessing for all friends' applications, or to determine the attributes that accessible by each friends' application individually. This technique will give the user more control of his information.

Further, in our scenario, we suppose that all application developers should inform the social network about the minimum attributes (necessary) for the application to provide the service. In addition, we suppose when a user registers at a social network, he/she decides either to give the same permission for all friends' applications or give the permission for each of them individually. If the user wants to give the permission of accessing to all friends' applications, then he/she should determine the general attributes to be accessed by friends' applications and the level of accessing as shown in Fig. 9.

4.3 Algorithm1 to access control on friends' applications

In the following algorithm the Social network: S is having:
Input : User set: $U=\{u_i \text{ where } i=1..\ .n\}$
Application set: $P=\{P_i \text{ where } i=1..\ .n\}$
User Data : A $=\{D_i \text{ where } i=1..\ .n\}$(age, birthday, photo)
Minimum attribute for application : Pi (mD1)
Boolean All_appction_access
General attributes access_by friends' applications: A ={GD1 , GD2, GD3}
Level of access L {Level 1 , level 2 }
Output : attributes if is allowed to access or notification
1) application P_i ⟶ requst_to_S_ to access_userAttribute (frindI , Pi,D1,D2)
2) S check If All_application_access =true
{
3) if level =1
// check the minimum attribute for Pi match with general attributes
4) if MD1=GD1.
5) Then Allow(mD1)
6) else return " This attribute not allowed to access by friends' applications "

7)If level =2
//check is this attribute is general
8)Then Allow(GD1)
}
9)Else if
10) Massage appear to user from S select the attribute you want to access by this friends' application
11) Return Attribute that allowed to access by this friends' applications.

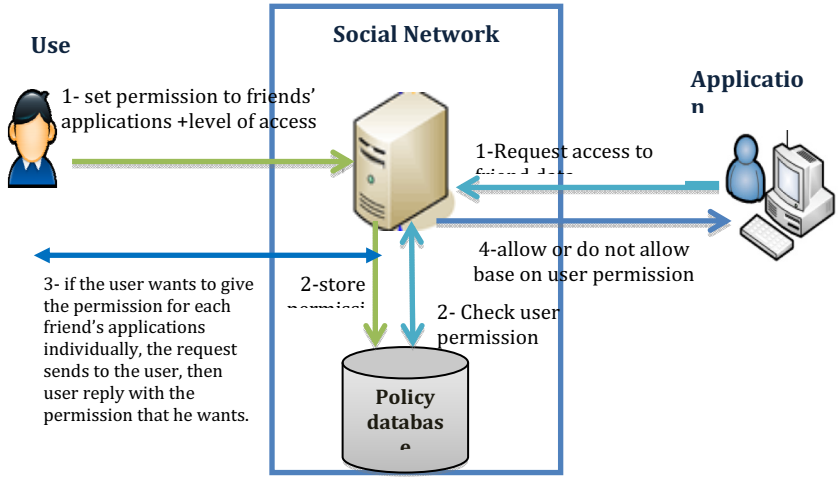


Fig. 9. Architecture for proposed solution – Part. 2.

As shown in Fig. 9 above, the proposed processes that the user should follow when he/she start registering in a social network are as follows:

1. User will determine the level to set the permission, as shown in Fig. 10.

Fig. 10. Determine the level of setting the permission for friends' applications.

1.1. Set the permission for all friends' application: means user will not specify different permission for each application. However, to protect user privacy, he/she should determine the following:

1.1.1. User determines general attributes to be accessing by friends' applications, as shown in Fig. 11.

Fig. 11. Determine general attributes to be accessed by friends' applications.

1.1.2. User gives restrictions in access control for friends' applications by giving the user two options, as shown in Fig. 12.

1.1.2.1. Allow friends' application accessing only necessary information if they match the policy determined before.

1.1.2.2. Allow friends' application accessing information determined before.

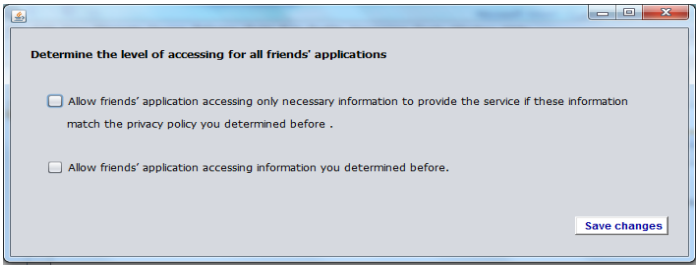


Fig. 12. Determine the level of accessing by friends' applications.

1.2. Set the permission for each friend's application individually as shown in Fig. 13. This gives the user ability to give the application permission to access only to attributes that necessary to provide the service. Therefore, the social network will notifies the user which attribute are necessary for this applications to provide the service.

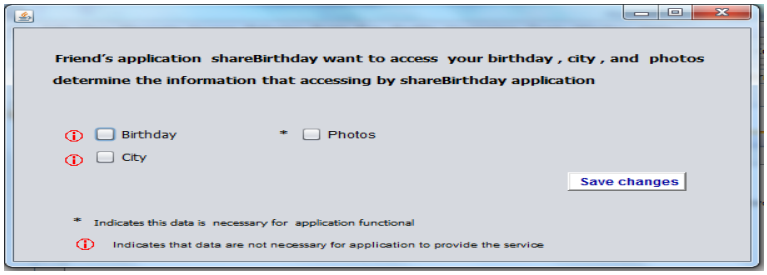


Fig. 13. Determine the attributes that accessing by each friends' application individually.

2. The determined permission by users will be stored in policy database of the social network.

In addition, when friends' application wants to access user data, the following process will be performed:

1. The friends' application send request to social network to access the user's data.
2. Social network checks policy database to know whether the user gives the permission for this application to access these data or not. If the user gives the permission for each friends' applications individually, go to step 3 then 4, else go to step 4 directly.
3. The request is sent to the user, then the user replies with the permission that he/she likes.

4. The social network will inform the application if it does not have a permission to access the data, otherwise the access request will be performed successfully.

5. Prototype Implementation

We evaluated the feasibility of our solution by simulating the social network environment through implementing a client server application by java language. Then, we connected the server with the database that we created to store data and policies for both the users and applications.
For problem.1, we simulate the social network control process of sharing data with friends under users policies. As shown in Fig.14, when application requests sharing user data with friends, the "Check_Sharing_Permission" method will be called and received the user id, application id, friend list and the data that app request to share them with user's friends. Therefore, server (social network) will check the database that contains the user policies to allow or prevent sharing request. If application has a sharing permission, the sharing process with allowed friend will perform successfully. Otherwise, the application will not be able to share this data because user does not give permission to this application to access or share.

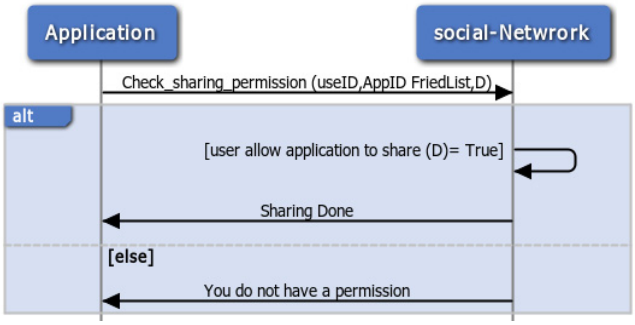


Fig. 14. Sequence of sharing data with friends.

For problem.2, we simulate the social network access control process for accessing data by fiend's applications under users policy after we implemented the algorithm. 1.
When a friend's application intends to access some attribute for users, it will send a request to a social network by calling "request_access_userAttribute" method, which informs the social

network the user id, application id, and the data that app request to access them, as shown in Fig. 15. Then the social network checks whether the user sets the permission to all friends' applications or not.

- If user gives the permission to each friends' application individually, a user will receive notification from social network by calling "request_access_attribute" method, that provides the user with the id of this application, data that application needs to access, and which attributes are necessary for this application to become functional, as shown in Fig. 15. Then the user determines the attribute to allow for this application to access, and send the permission to social network, which allow to application to access attribute depended in this permission.
- If the user gives the permission to all friends' applications at once. Then the social network checks the level of accessing. If it is level 1, the social network knows that the user allows only the minimum attribute of the application. If it is level 2, social network knows that the user allows the general attributes that defined already to access with friends' applications and allow the application to access permitted attributes.

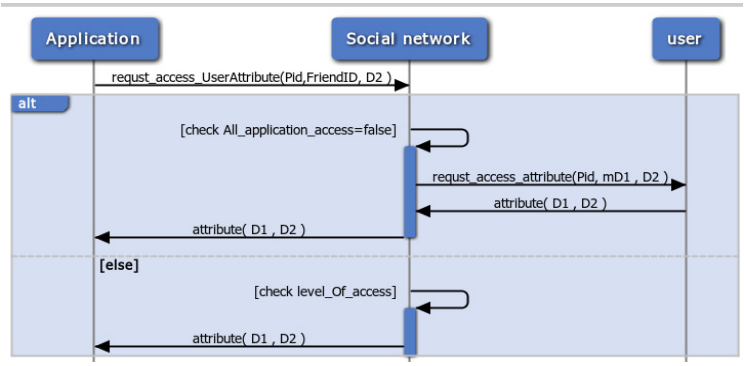


Fig. 15. Sequence diagram for Algorithm. 1.

6. Results and Discussion

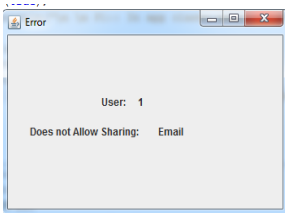
In this section, we focus on the evaluation of our hypothesis, to know whether they are valid or not. In our client-server application we run both applications where the user considers as client and the server act as social network that have the database. Then, we test different users

inputs in order to check their effect on application side. We provided the print screens for both user and application.

6.1 For problem 1

We tried to test two cases for two users. We suppose both of them installed application called "share birthday" which needs two attributes (birthday and email) to be accessed. Further, we consider that the users want to use this application to share their birthday with friends so both of them give the application permission to "access and share" their birthday. However, for 'email' attribute, user.1 allows access whereas user.2 allows application to access and share with friends as shown in Fig.16(a), Fig.16(b) and Table 1. For user.2 when application invokes Check_sharing_permission method, which is explained in section 4, the social network checks the roles table in database and finds that the user gives a permission to share this attribute. So, sharing done successfully as shown in Fig.17(b). However, for user.1 sharing will not be allowed as shown in Fig. 16, because when the social network checks the roles table, it finds the application has a permission to access only, as shown in Fig.17(a).

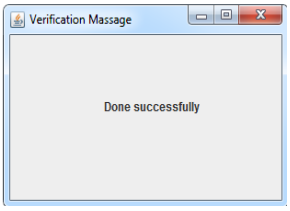
(a) User.1 input to 'Roles database'



(b) output massage for application

Fig. 16. Result for user.1 test.

(a) User.2 input to 'Roles database'



(b) output massage for application related to user.2

Fig. 17. Result for user.2 test.

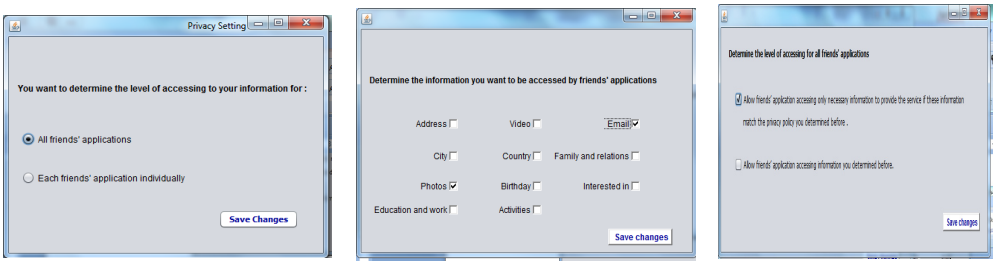
Table 1. Users level of permission for ShareBirthday application.

User number	Name of attribute (D)	Level of permission	Social Network - process
1	'Birthday'	Access and share	Allow sharing
2	'Birthday'	Access and share	Allow sharing
1	'Email'	Access and share	Allow sharing
2	'Email'	Access only	Do not Allow sharing

Therefore, we found that these techniques will increase users control on data to protect privacy and empower users to prevent sharing data with friends without permission. This means that the possibility of violating their privacy by a third-party application decreases.

6.2 For problem 2

We suppose there are three users, and an application called "share photos" needs an email and photos of the user, while its minimum attribute is photos. For user.1 he/she gives the permission to all friends' application to access only the minimum attributes it needs when they match with general attributes as shown in Fig.18 (a, c). The user sets the general attributes as follows: email and photos, as shown in Fig.18(b). When "share photos" application requests to access user.1 data to test the permission for photos and email attributes, the result is the photos access by "share photos" application as shown in Fig.19(a).



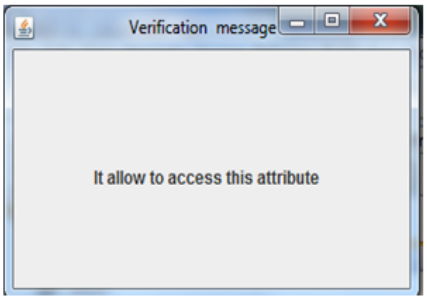
A) User.1 determines the level of setting the permission for friends' applications

b) User.1 determines general attributes to be accessed by

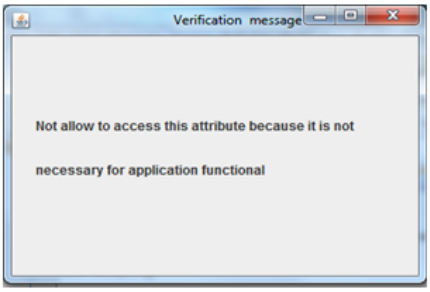
c) User.1 determines the level of accessing by friends' applications

Fig.18 User.1 input to 'Roles database'.

Because user.1 determines the minimum attributes should be accessible by friends' application, when they match the general attributes, the application cannot access to email, because it is not necessary for its function, as shown in Fig.19 (b).



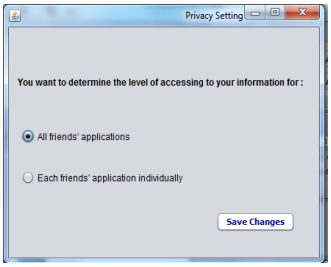
a) The result for test user.1 photos attribute



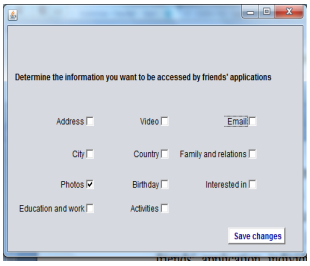
b) The result for test user.1 email attribute

Fig. 19. Result for user.1 test.

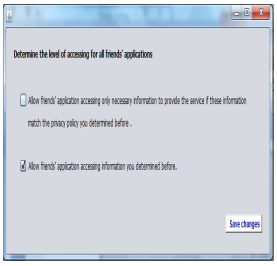
User.2 gives the permission to all friends' applications to access the general attributes and determines the photos as general attributes as shown in Fig.20. The result is the photos access by "share photos" application, but the application cannot access to email because user.2 doesn't determine it as general attributes, as shown in Fig.21.



A) User.2 determines the level of setting the permission for friends' applications

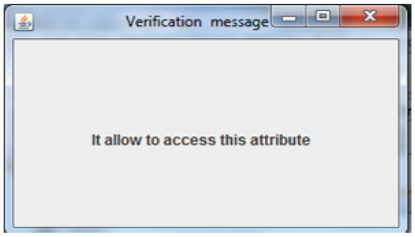


b) User.2 determines general attributes to be accessed by friends' applications

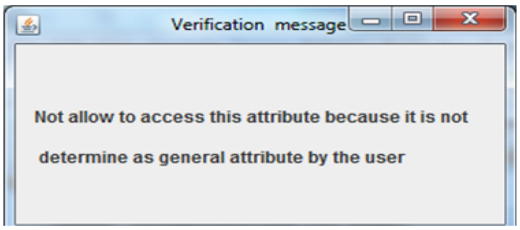


c) User.2 determines the level of accessing by friends' applications

Fig. 20. User.2 input to 'Roles database'.



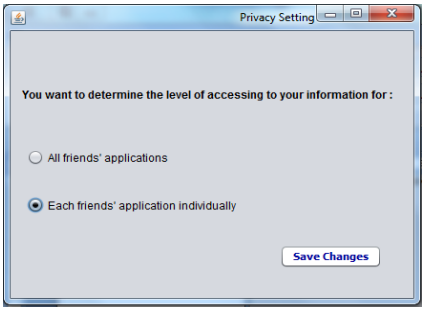
a) The result for test user.2 photos attribute



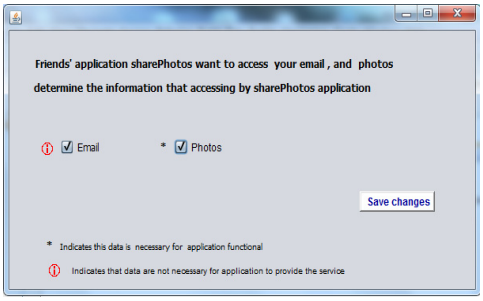
b) The result for test user.2 email attribute

Fig. 21. Result for user.2 test.

User.3 wants to determine the level of accessing for each friends' application individually as shown in Fig.22(a). For example, "Share Photos" application want to access user.3 photos, email. The result will depends on the user's permission as shown in Fig.22(b). Table.2 summarizes these three cases. These results indicate that our hypothesis is correct.



a) User.3 determines the level of setting the permission for friends' applications



b) User.3 determines attributes to be accessed by share Photos applications

Fig. 22. Result for user.3 test.

Table 3. Summary of three cases.

User ID	Permission to all friends' application	Level of accessing by friends' application	General attributes to be access by friends' application	Minimum attribute for "share photos" application	Unnecessary attribute for share photo application"	Accessing attribute by "share photos" application
1	True	Level 1	Email, photos	Photos	Email	Photos
2	True	Level 2	Photos	Photos	Email	Photos
3	False	-	-	Photos	Email	Photos, email

7. Conclusion

Recently, the concern about user privacy in social network is raised, because the number of users who use the online social network to share personal and private information is increased. Furthermore, third-party applications in social network are becoming a major type of online applications as they provide more attractive services for users. However, we cannot trust those applications developers since their intents are not known. Therefore, preserving user privacy in social network from third party applications has become an important concern. This paper focused on this issue and discussed two problems, which are, how could users avoid sharing their private sensitive data with their friends without their permission, and how could they customize the access of friend's applications to their data.

To avoid leaking personal information to friends, the authors suggest giving users, through installation process, more ability to specify level of permission for each data asked by third-party application. In addition, to give users control on friend's application, the authors propose an algorithm to enable them determining the attributes that can be accessed. This determination can be for all friends' applications at once or for each application individually. Moreover, to evaluate the feasibility of the proposed solution, the authors simulate the environment of social network by building client server application by java language. Then, they connected it with a database containing applications and users data and policies.

We found that when user applies our approach, he/she can change the permission for each attributes and become able to achieve for his data the level of privacy that he/she wants. Therefore, the researchers found that their proposed solutions help users to control their data given to

third-party application, especially the sensitive and private information that they usually keep private or visible only to specific group. Overall, this work focuses on preventing third-party application from sharing user's data with friends in addition to avoiding friend applications from getting data without permission. However, there is still a need for future researches to find how to prevent third party applications from exposing publicly or to advertisers, the information that we allow to them.

References

[1] **Cutillo, L. A., Molva, R.** and **Strufe, T.,** "Safebook: A privacy-preserving online social network leveraging on real-life trust", *Communications Magazine, IEEE*, 47.12: 94-101, (2009).

[2] **Tripathy, B. K.** and **Panda, G. K.,** "A new approach to manage security against neighborhood attacks in social networks", *Advances in Social Networks Analysis and Mining (ASONAM)*, *International Conference on. IEEE*, (2010).

[3] **Masoumzadeh, A.** and **Joshi, J.,** "Osnac: An ontology-based access control model for social networking systems." *Social Computing (SocialCom), IEEE Second International Conference on. IEEE*, (2010).

[4] **Adrienne, F.** and **Evans, D.,** "Privacy protection for social networking APIs", *Web 2.0 Security and Privacy* (W2SP'08) (2008).

[5] **Sean, C.** and **Srivastava, G.,** "Social network privacy for attribute disclosure attacks." *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE, (2011).

[6] **Heng, X., Wang, N.** and **Grossklags, J.,** "Privacy By Redesign: Alleviating Privacy Concerns For Third-Party Applications, (2012).

[7] **Shehab, M., *et al.*,** "Access control for online social networks third party applications." *Computers & Security* (2012).

[8] **Bimal, V., Kiciman, E.** and **Saroiu, S.,** "Keeping information safe from social networking apps." *Proceedings of the 2012 ACM workshop on Workshop on online social networks*. ACM, (2012).

حماية خصوصية المستخدم من تطبيقات شبكات التواصل الاجتماعية

إبتسام العمري، ومرام الحافزي، وعمر باطرفي

قسم تقنية المعلومات

كلية الحاسبات وتقنية المعلومات

جامعة الملك عبدالعزيز، صندوق بريد ٨٠٢٢١، جدة ٢١٥٨٩،

المملكة العربية السعودية

obatarfi@kau.edu.sa

المستخلص. يومًا بعد يوم، يزيد عدد مستخدمي شبكة التواصل الاجتماعي وبالتالي يزيد استخدام التطبيقات المختلفة التي توفرها شبكات التواصل لنشر المعلومات الشخصية . علاوة على ذلك، فإن المستخدمين يحرصون على حماية خصوصياتهم ومشاركة هذه الخصوصيات مع أفراد العائلة أو الأصدقاء. ومع ذلك، فإن العديد منهم غير مدركين لدرجة هذه الخصوصية. وفي هذا البحث، سيتم استعراض خصوصيات المستخدمين لإنشاء التطبيقات، والتركيز على منهجية منع مشاركة الخصوصية بين الأصدقاء داخل شبكة التواصل دون إذن المستخدم. وأخيراً سيتم تطبيق آلية تقويم النظام المقترح لجدوى النظرية مع إجراء نظام محاكاة بلغة جافا في بيئة الخادم والعميل.