Analyzing Privacy Designs of Mobile Social Networking Applications

Guanling Chen and Faruq Rahman
Department of Computer Science, University of Massachusetts Lowell
{glchen, frahman}@cs.uml.edu

Abstract

The combined advances of open mobile platforms and online social networking applications (SNAs) are driving pervasive computing to the real-world users, as the mobile SNAs are expected to revolutionize wireless application industry. While sharing location through mobile SNAs is useful for information access and user interactions, privacy issues must be addressed at the design levels of mobile SNAs. In this paper, we survey mobile SNAs available today and we analyze their privacy designs using feedback and control framework on information capture, construction, accessibility, and purposes. Our analysis results suggest that today's mobile SNAs need better privacy protection on construction and accessibility, to handle increasingly popular mash-ups between different SNA sites. We also identify two unexpected privacy breaches and suggest three potential location misuse scenarios using mobile SNAs.

1 Introduction

Recent advances on capable mobile devices and social networking applications (SNAs) are quickly converging, accelerating the transition of pervasive computing from vision to reality. The open mobile platforms, particularly Apple iPhone and Google Android, make it much easier than before for developers to build third-party applications that may potentially used by millions of people on their always-on always-carried mobile devices. While Google Android is yet to be released, Apple iPhone has already claimed six millions of users and expects to sell more than 24 million units in 2009 [4].

On the other front, online SNAs, such as Facebook and MySpace, have become extremely popular in the past several years. For example, Facebook had 123.9 million unique visitors in May, 2008 [12]. Given the availability of open mobile platforms, it is only natural to expect that people will increasingly use SNAs on their cellphones. In particular, iPhone has unique multi-touch interface, geo-localization capability, and embedded sensors, which may well boost user experience of mobile SNAs.

As location can be used to find and interact with nearby events, business, and friends, privacy concerns remain as a significant design challenge for mobile SNAs. There have been several user studies on privacy issues of location disclosure [1, 14, 3, 8, 11] and several guidelines on protecting privacy have been proposed [2, 7, 5, 6, 9, 10]. It is, however, unclear how real-world applications, particularly mobile SNAs that leverage location, have implemented privacy protections.

In this paper, we analyze the privacy designs of 31 mobile SNAs listed in Apple App Store, available for free to millions of iPhone users. We use Bellotti and Sellen's *feedback* and *control* framework [2] for this study. We found that the privacy designs for information construction and accessibility are particularly weak for many mobile SNAs, and we identified two unexpected privacy violations and suggest three misuse scenarios. A fundamental reason that causes these issues is the popular "mash-ups" of different SNA sites. Users have little feedback and coarse control on the information flow among these sites, which can be particularly problematic since users may have different sets of friends and inconsistent privacy policies.

To the best of our knowledge, this privacy study of real-world mobile SNAs is the first of its kind. While this paper focuses on an informal framework, it lays out a context for any further formal study. The rest of this paper is organized as follows. Section 2 describes the mobile SNAs we studied and we present analysis results in Section 3. We discuss related work in Section 4 and conclude in Section 5.

2 Mobile Social Networking Applications

The defining feature of Web 2.0 applications is the user-generated content, which is used to facilitate *information access* and *user interactions*. The content shared by users could be many different types of information, such as videos (YouTube), photos (Flickr), Web pages (Del.icio.us), or status updates (Twitter). One may, how-

ever, differentiate SNAs with the traditional Web 2.0 applications as the SNAs allow a user to define a set of *friends*, whose activities are automatically visible to that user. For example, Amazon allows users to review products but no friendship among users can be defined. On the other hand, Facebook has an explicit friendship circle defined by individual users who get automatic updates on their friends' activities. Sharing through friends gives users incentives to return and enables viral growth of SNAs' user populations.

But what applications can be counted as *mobile* SNAs? To answer this question, we studied 31 applications listed in the "Social Networking" category of the Apple App Store (as of July 26, 2008 – two weeks after the opening of the App Store). These applications are all free and run on Apple iPhone (or iPod Touch). We classify these applications into four groups, as shown in Table 1, and compare them based on whether they use location, whether they allow users to define friends, and whether they allow users to interact with nearby strangers (non-friends).

The *mobile frontends* are mobile representations of their desktop counterparts, such as instant messengers, or well-established SNA sites, such as MySpace and Facebook. They typically have well defined friendship and do not explicitly support the interactions between non-friends. While most of them have not added location support at this time, it is likely that this feature will be added in the near future.

The *content sharing* applications allow users to capture and upload text, photo, voice, and video messages to a variety of SNA sites. For example, ShoZu can upload photos to more than 40 sites, such as Flickr and Facebook. Recently *microblogging* applications have become quite popular; they allow users to write and publish brief text *updates*, either to be viewed by anyone or only by permitted *followers*. Updates from people a user follows will be automatically received by that user. The most popular microblogging service is Twitter, while many other sites (such as Facebook) has also implemented this feature through "status updates." The last 6 applications in this group (Table 1) are microblogging services, with the PhotoShare focuses on photos and the rest focuses on text (though it is possible to share text links of various media content). Both Exposure and Twinkle allow users to browse and comment on photo and text updates from nearby non-friend users. Like mobile frontends, these content sharing applications are often augmented extensions to existing Internet sites.

Some SNAs are designed to allow users to make new friends. The *neighborhood exploring* applications allow users to leave text, photos, scribbles, or voice remarks on "virtual walls" at certain locations; and these walls can be discovered and read by nearby users. All these applications rely heavily on location and anonymized interactions. The "Eventful" application allows users to find and comment on nearby upcoming events, and also to leave remarks on other users' profiles, which are presumably discovered through comments on mutually-interested local events. Users who become more friendly through these interactions may choose to exchange their contact information and meet in real life.

The *mobile-specific SNAs* are designed specifically for mobile community. Whrrl, Loopt, and Limbo all allow users to see their friends' locations, activities, and their comments about places. Avatar and Bluepulse have not used location and focus on gaming community and SMS/email communications, respectively.

	Location	Friendship	Nearby
Mobile frontends			
AIM	No	Yes	No
Palringo	No	Yes	No
MySpace	No	Yes	No
Facebook	No	Yes	No
CenceMe	Yes	Yes	No
mDialog	No	Yes	No
Content sharing			
Kyte	No	No	No
Typepad	No	No	No
CellSpin	No	No	No
Lifecast	Yes	No	No
SodaSnap	Yes	No	No
Plum	No	Yes	No
ShoZu	Yes	Yes	No
Exposure	Yes	Yes	Yes
PhotoShare	No	Yes	No
Pownce	No	Yes	No
Twinkle	Yes	Yes	Yes
Twittervision	Yes	Yes	No
Twittelator	Yes	Yes	No
Twitterrfic	Yes	Yes	No
Neighborhood exploring			
Graffitio	Yes	No	Yes
zintin	Yes	No	Yes
WhosHere	Yes	No	Yes
GeoGraffiti	Yes	No	Yes
iFob	Yes	No	Yes
Eventful	Yes	No	Yes
Mobile-specific SNAs			
Whrrl	Yes	Yes	No
Loopt	Yes	Yes	No
Limbo	Yes	Yes	No
Avatar	No	Yes	No
Bluepulse	No	Yes	No

Table 1. A list of SNAs in Apple App Store for iPhone (as of July 26, 2008).

Out of these 31 applications, 18 of them use location to find nearby business, events, friends, and other users' comments; 20 applications allow users to directly interact with their friends on the mobiles; and 8 applications allow spontaneous close-by interactions between non-friends. It is clear that location and friendship are important for mobile SNAs; only 3 of the 31 applications use neither of these two features.

3 Analysis of Privacy Designs

We analyze the privacy designs of mobile SNAs using Bellotti and Sellen's *feedback* and *control* framework [2]. This framework considers four components regarding information flow: 1) *capture*: what kind of information is being collected? 2) *construction*: what happens to user's information once it is collected? 3) *accessibility*: who can access the collected information; and 4) *purposes*: how is the information used by other people? The framework allows us to analyze what feedback and control an application provides along these four aspects. Our discussions are focused on user's location, the most important information for mobile SNAs.

3.1 Capture

The majority of iPhone mobile SNAs we surveyed use a popup dialog to ask for permission to acquire current location (shown in Figure 1). This feedback mechanism lets users know when their location is captured and gives users full control whether to grant this request. Loopt, Graffitio, and Twinkle, however, seem to automatically acquire location at startup with a short message showing on the status bar. Users thus have feedback but no control to disallow location capture.

There is little feedback and control provided by mobile SNAs on whether the location information is *continuously* acquired. We know that Loopt requires continuous location updates, based on the feedback of periodic "Locating..." messages on the status bar. The user cannot control how frequently, when, and where the location can be continuously acquired. Rather, the only control Loopt provides is to disable location updating all together.

Instead of automatic location acquisition, some applications require user to take explicit actions. For example, Twittelator users need to click a button if they want to include their current location in the status update. BrightKite[1] requires users to manually supply current location (BrightKite is a Web application and thus is not listed in Apple App Store and Table 1). Similarly,

[1] http://brightkite.com/

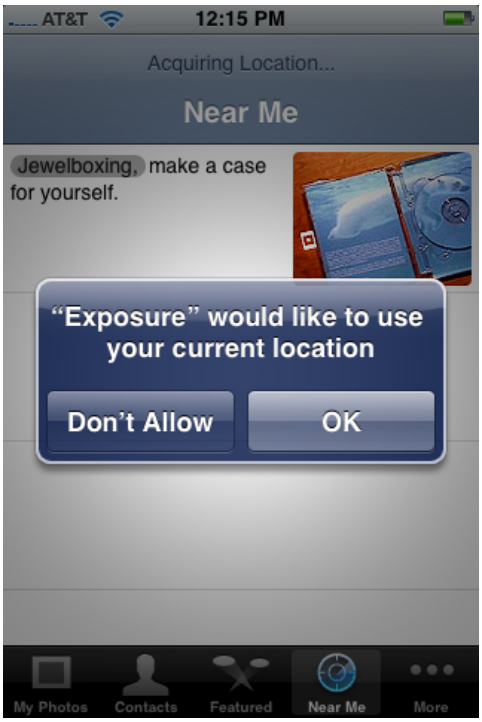

Figure 1. Exposure asking for localization permission.

Loopt also allows users to manually input location if its automatic location updating is disabled.

The accuracy of location depends on mobiles' capability and whether they are indoors or outdoors. The first-generation iPhones use both cellular signal triangulation and WiFi signal databases to find location, while the iPhone 3G uses GPS that can give much accurate location outdoors. Some applications do not allow users to change the location granularity. For example, Twittelator actually publishes coordinates that can be accurate to several meters. On the other hand, BrightKite and Loopt (in manual mode) allows users to control the accuracy of their location visible to others.

It appears that existing mobile SNAs have various feedback and control mechanisms over capturing user location, though most location acquisition policies are quite simple. Some balances are necessary between users having full control and harassing users to input location frequently on a small device. We suggest that better feedback on continuous location collection and better control over location granularity should be considered for improvements of existing mobile SNAs.

Besides location, there is no obvious feedback and control on whether users' other information, such as the identity, phone number, calendar, contact list, and call history, is implicitly collected by these applications. It is

particularly worrisome since some applications are written by independent (and maybe anonymous) developers. While Apple may perform some sanity checks before accepting and distributing these applications through App Store, users have to put great trust by running third-party applications on their personal devices.

3.2 Construction

What happens to users' information once it is collected depends greatly on individual applications. For *neighborhood exploring* applications, it is reasonable to assume that user's location will be sent back to a server from which updates of nearby users can be downloaded. But is the location also cached locally? Is it sent over to the server using encrypted connections? Is it stored at the server, and for how long? Will it be shared with third parties? Unfortunately for most applications, there is no or little feedback and control once personal information gets into the system.

Information flow becomes more complicated and subtle as more SNA sites are mashed up together. Namely, an update on one site will be automatically published on another site, if a user has profiles on both sites and chooses to set up this link. For example, a video marked as favorite on YouTube may get published on FriendFeed, and then pushed further to Facebook. Many of the mobile SNAs listed in Table 1, such as CellSpin, LifeCast, ShoZu, Twinkle, and Loopt can easily link to Twitter, a popular microblogging service.

Consider the Twitter example a bit further (Twittelator and Twitterrfic are iPhone clients for Twitter). From its website, it is clear that every update is archived in Twitter's databases. If a user does not protect her updates, they will also appear on the "public timeline" that is visible to everyone. Twitter also has APIs allowing third parties to retrieve the public timeline, thus a user may never know where her updates eventually reach. For example, Summize[2] archives Twitter's public timeline messages and make them globally searchable. Twitter updates may also be pushed to friends through XMPP messaging service, thus the XMPP server in the middle can easily intercept and store the updates. There is no or little feedback on these external information flows.

If a user later chooses to delete some of her updates on Twitter, the messages still remain in third-party repository, such as in Summize's databases, and are likely to be still publicly available. Thus the user's control of *message deletion* is limited on Twitter. As the time of this writing, Summize is acquired by Twitter, though their databases appear to remain separate. A user may choose to protect her updates through the preference option, so her messages are only available to her friends. Her friends may use APIs to easily archive all messages and may even rebroadcast her updates to the public timeline (called "retweet"). Thus the user's control of *message protection* is also limited on Twitter.

Some user interface issues, because of lacking feedback, provide further confusions to where the location information goes. For example, clicking the location button when posting updates on Twittelator will insert a shortened link to Google Maps of current location to the message. On the other hand, clicking the location button when using Twitterrific will actually automatically change the location of user's profile on Twitter without any visual confirmation.

In summary, the feedback and control designs are weak in many mobile SNAs and may become even worse as SNA sites are increasingly mashed up. While these issues are not specific to mobile SNAs, the use of sensitive location information pose greater privacy threats if these issues are not addressed appropriately. We believe that SNAs need to provide better feedback and control, while users also need to be responsible on setting up the automatic "pipes" between the SNA sites.

3.3 Accessibility

For neighborhood exploring applications, location information should only be kept at and accessible by the service providers. A user may be discovered by others as "nearby," but the exaction location (and often identity) should never be shared with non-friends. This access model is usually understood by the users, though no explicit feedback is provided by most applications. In almost all cases, users do not have control over the distance between those who can discover them.

For mobile-specific SNAs, existing applications all provide users full control on who can access their current location. No feedback, however, is given to users on who have actually viewed their location at what time. This arguably can be considered as privacy protection for those who checked users' location, despite of that researchers have argued to minimize asymmetric information flow [6].

For content-sharing applications, accessibility becomes difficult to track as user's updates propagate through various SNA sites, on which the user may have a set of different friends and thus different access policies. We give two examples of unwanted location exposures for Twitter users who protect their updates (only viewable to their Twitter friends). When posting through Twitterrfic, users can click location button that will automatically update the location of their profiles to be users' current location, such as "Location iPhone: 45.488113,-90.578766." The coordinates can easily be located by searching Google Maps. Thus a user's location is leaked

[2]http://www.summize.com/

through her profile, which is publicly viewable even if her updates are protected.

The other example of location leakage is caused by using Twittelator to publish a photo and attach current location to it. Since Twitter only allows text updates, the photo will be uploaded to TwitPic[3] and a link to that photo is published on Twitter together with another link to a Google Map of current location. Unfortunately TwitPic makes everything public, while the user may think her updates are only available to her friends. In both cases, Twitterrific and Twittelator, users have no feedback and control on these privacy violations.

Due the popularity and ease of use of microblogging services, Twitter has also emerged as a messaging platform that may have subtle implications on conversational privacy. For example, the conversation between two users using update-and-reply is visible to their mutual friends, which may not be the intended consequence. If only one user has protected her updates, the other half conversation will appear on public timeline, making it possible to guess the protected messages based on the conversational context.

In summary, like construction, inconsistent policies of linked SNA sites make accessibility difficult to track. This may result in both explicit and subtle privacy risks, which may become particularly dangerous when location and identity are leaked, since usually no feedback and control mechanisms are given to the users.

3.4 Purposes

As Bellotti and Sellen point out, why other people access our personal information is outside of the system [2]. It may only be possible, but not guaranteed, to infer purposes from construction and access patterns. Users can only exercise social controls to restrict unethical and illegal usage of their personal information.

Here we give three examples of potentially unwanted interactions by using mobile SNAs. First consider a simple example using neighborhood exploring applications, some of which allow nearby users to post comments and photos on each other's "walls" anonymously. More than one users, however, have reported that pornography content were posted to their walls only hours after their zintin/PhotoShare walls were established. While the purposes of the offenders remain unclear, this practice is extremely annoying and may turn users away from using such mobile SNAs. The feedback here is the actual content on users' walls, and users may take control to delete or report abuse to application providers.

The second example may show some unexpected revelation of a user's true identity. For example, a user may choose to set up an anonymous profile on Flickr and publish beautiful and funny photos, which do not contain identity-related information. On the other hand, people in her region may discover her photos using Exposure, and may recognize the photos either because she has shown to them or they may realize the content/context of the photos. Thus the Flickr user's true identity may be revealed because the photos serve as the link between her virtual and real social networks. To make things worse, most users use the same login name across various SNA sites [15], thus the complete anonymous social life of a user may be exposed to her friends and families. Though this may also happen without using Exposure, the nearby search functionality certainly makes the linkage much easier to discover.

Finally the history of location information may reveal more sensitive information about a user, particularly when data mining based automated methods are used. For example, we extracted a user's Twitter updates that contain location published through Twittelator, over the past two weeks since Twittelator becomes available on iPhone. There are 12 such updates and we plotted them on Google Maps, shown in Figure 2. The homework two-cluster pattern becomes immediately visible, without using any other tools. That user reported 4 updates in Los Angeles, then 6 updates in San Francisco area, and then 2 updates back to Los Angeles. It may seem to be odd since the distance between the two clusters is quite large for most commuters. We did, however, confirm through the content of that user's updates that this person is a remote worker, each week spending several consecutive days at work and home, respectively. While one may argue that a vacation trip may also result in a similar pattern, we believe that such geographical and temporal analysis of a longer-time location history will inevitably pose significant privacy threats.

In summary, it is difficult to control how personal information is used once it has become available. The providers of mobile SNAs must consider limiting information construction and auditing information accessibility from the beginning of application designs.

4 Related Work

Privacy protection in pervasive computing is an important subject and has been well researched. Researchers have generally conducted two types of privacy studies: one is to construct risk models and provide guidelines on good privacy designs [2, 7, 5, 6, 9, 10], and the other is to conduct user studies with real applications [1, 14, 3, 8, 11, 13]. Both provide helpful insights on privacy designs, though most of existing work has focused on small-scale academic research applications. In this paper, we study existing (commercial) mobile

[3] http://www.twitpic.com/

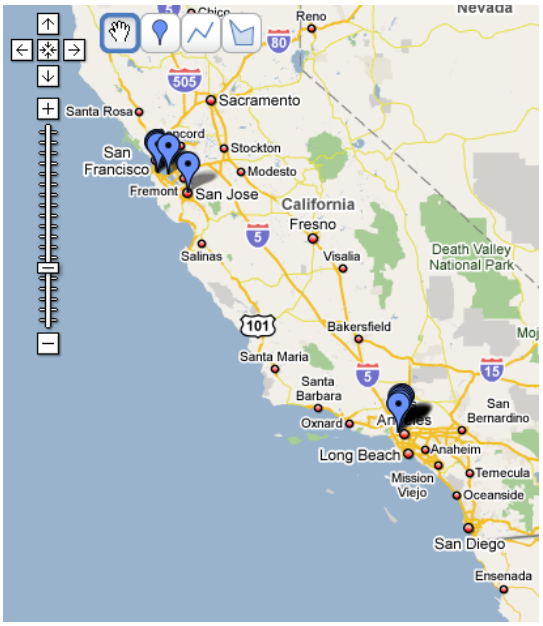

Figure 2. The apparent home-work location clusters from a Twitter user.

SNAs provided by developers, instead of researchers, and we show the gaps between suggested models and actual practices for privacy protection issues. Hsieh et al. have designed their instant messaging application using Bellotti and Sellens feedback and control framework [2], while our focus is to use this framework to evaluate location privacy of existing mobile SNAs, rather than building our own applications.

5 Conclusion

The analysis of the privacy designs for existing mobile SNAs suggests that both feedback and control of information construction and accessibility are weak for existing applications. A particular problem is automatic mash-ups between various SNA sites, which expose personal information flow to multiple entities and inconsistent access policies may result in privacy breaches, as we identified two such cases. In the future work, we plan to conduct user studies using real-world mobile SNAs and make specific suggestions on how to mitigate privacy concerns at application design levels.

References

[1] L. Barkhuus and A. Dey. Location-based services for mobile telephony: a study of users' privacy concerns. In *Proceedings of the 9TH IFIP TC13 International Conference on Human-Computer Interaction*, July 2003.

[2] V. Bellotti and A. Sellen. Design for privacy in ubiquitous computing environments. In *Proceedings of the Third Conference on European Conference on Computer-Supported Cooperative Work*, pages 77–92, Milan, Italy, 1993.

[3] S. Consolvo, I. E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge. Location disclosure to social relations: why, when, & what people want to share. In *Proceedings of the 2005 ACM Conference on Human Factors in Computing Systems*, pages 81–90, Oregon, PL, Apr. 2005.

[4] D. Frommer. Apple's iPhone 3G is the new iPod, sales to triple. Silicon Alley Insider, June 2008.

[5] J. I. Hong, J. D. Ng, S. Lederer, and J. A. Landay. Privacy risk models for designing privacy-sensitive ubiquitous computing systems. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, pages 91–100, Cambridge, MA, 2004.

[6] X. Jiang, J. I. Hong, and J. A. Landay. Approximate information flows: Socially-based modeling of privacy in ubiquitous computing. In *Proceedings of the International Conference on Ubiquitous Computing*, pages 176–193, Göteborg, Sweden, 2002.

[7] X. Jiang and J. A. Landay. Modeling privacy control in context-aware systems. *IEEE Pervasive Computing*, 1(3):59–63, 2002.

[8] A. Khalil and K. Connelly. Context-aware telephony: Privacy preferences and sharing patterns. In *Proceedings of the 20th Conference on Computer Supported Cooperative Work*, pages 469–478, 2006.

[9] M. Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. In *Proceedings of the International Conference on Ubiquitous Computing*, pages 273–291, Atlanta, GA, 2001.

[10] S. Lederer, J. I. Hong, A. K. Dey, and J. A. Landay. Personal privacy through understanding and action: Five pitfalls for designers. *Personal and Ubiquitous Computing*, 8(6), Nov. 2004.

[11] S. Lederer, J. Mankoff, and A. K. Dey. Who wants to know what when? privacy preference determinants in ubiquitous computing. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 724–725, Ft. Lauderdale, FL, 2003.

[12] S. Olsen. Facebook's Sandberg: Growth before monetization. News.com, July 2008.

[13] M. Prabaker, J. Rao, I. Fette, P. Kelley, L. Cranor, J. Hong, and N. Sadeh. Understanding and capturing people's privacy policies in a people finder application. In *Proceedings of the Workshop on Ubicomp Privacy*, Innsbruck, Austria, Sept. 2007.

[14] I. E. Smith, S. Consolvo, A. LaMarca, J. Hightower, J. Scott, T. Sohn, J. Hughes, G. Iachello, and G. D. Abowd. Social Disclosure of Place: From Location Technology to Communication Practices. In *Proceedings of the Third International Conference on Pervasive Computing*, Munich, Germany, May 2005.

[15] M. N. Szomszor, I. Cantador, and H. Alani. Correlating user profiles from multiple folksonomies. In *Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia*, pages 33–42, Pittsburgh, PA, June 2008.