The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.41 *e-ISSN: 2251-7545*

Identity Hopping Based Security Framework for Social Media Applications

[*]Ahmad Zmily[1], Dirar Abu-Saymeh[2], Dhiah el Diehn I. Abou-Tair[3]
[1,2,3] *School of Information Technology and Engineering,*
German Jordanian University Amman, Jordan
Email: [1]ahmad.zmily@gju.edu.jo, [2]dirar.abusaymeh@gju.edu.jo, [3]dhiah.aboutair@gju.edu.jo

Abstract. **With the advent of mobile devices and social networks, information and identity security concerns have increased. Mobile devices that have multiple sensing capabilities have been interfaced with social networks allowing users to post many of their activities and habits instantly onto social networks. This information can easily be used by social network providers to invade privacy and pose security risks to users. In this paper we propose an identity security framework that encapsulates a generic interface between mobile devices and social networks that utilizes identity hopping to secure and hide users' real identities. The framework also employs anti-correlation measures to prevent social network providers from being able to correlate the identities together. The proposed framework has been implemented on the Google Latitude application as a case study to hide users' real identities and prevent the service provider from tracking complete movement habits. The implementation shows the effectiveness of the proposed interface in enhancing identity security.**
Keywords*: Identity Security; Hopping; Location Privacy; Security Framework; Social Media Applications.*

1. Introduction

In recent years, mobile phones have improved rapidly in processing power, embedded sensors, storage capacity, and network data rates. The mobile phones of today have evolved from merely being phones to full-fledged computing, sensing, and communication devices. These advances in mobile phone technology have paved the way for exciting new applications. At the end of 2011, there were 5.9 billion estimated mobile subscribers world wide [1] with more than 600,000 applications available for Apple's iPhone and more than 500,000 applications available for the Google Android platform [2].

Mobile phone applications typically have unrestricted access to personal information such as contacts, photos, text messages, user records, user location, and locally stored data. This unrestricted access posses a threat to users' security and increases the possibility of identity theft. The increasing growth in number of smart mobile devices coupled with widespread availability of applications is pushing security and privacy threats to new high levels.

With the advent of social networks and the integration of mobile devices with these networks, application service providers of such social networks now have widespread access to even larger correlated identity information. An application service provider can easily correlate a user's behavior with the behavior of other people that are part of the user's social network and derive information that the user has not even stored on the device. For example, the mere fact that many people of a user's social network frequently visit a specific place is a good indication that the user may also visit that location even if that user is not tracking his own movement on his mobile device. The knowledge, that a user belongs to a specific social network, now encapsulates a long list of identity information.

Application service providers are expected to set and honor privacy settings that prevent tracking and collecting users' personal activities and information. Nevertheless, many companies have been caught violating users' privacy settings. Just recently, several advertising companies including Google have been bypassing the privacy settings of millions of people using Apple web browsers on their iPhones and computers [3]. The companies used special computer code that tricks Apple's Safari web-browser into letting them monitor many users. Safari, the widely used browser on mobile devices, is designed to block such tracking by default.

Application service providers should not be solo trusted with protecting users' identities and security. Ensuring the security of a user's identity has to start with preventing an application service provider from knowing the user's real identity and the social networks or any kind of a group that the user belongs to. This all has to happen without sacrificing any of the benefits that such group associations have to offer a user and his social network.

In this paper, we propose a new architectural framework that can be used to hide users' real identities in social networks that require users to share their personal information including location without any sacrifice to users' ability to fully utilize the offered services. Our

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.41 *e-ISSN: 2251-7545*

proposal uses a generic identity hopping technique to hide real identities. The framework dynamically switches between identities while users are connected to social networks. We have also developed an algorithm to optimize switching between the different identities. A synchronization process is used to ensure that other users of social networks can correlate the activities of the various identities to come up with a single user view.

To illustrate the effectiveness of the proposed framework, we have implemented it for Latitude, a Google mobile application that tracks users' location and movement. The case study shows how the framework can be used in a real application to secure users' identities and how the framework parameters are chosen based on the application domain.

The remainder of this paper is structured as follows. In Section 2, we describe in details the solution framework architecture and the generic algorithms that have been developed. Section 3 presents the application of this framework to the Google Latitude application. In Section 4, we discuss related work and describe how the proposed framework differs from previous related activities. We conclude with final comments in Section 5.

2. Identity security framework for social networks

Social Networks pose a large threat to privacy and security since users inherently upload much of their personal information into the social network sites. To enhance privacy and security in such environments, a framework has been developed as depicted in Figure 1. The framework utilizes multiple identities to enhance privacy and security. A user is assigned an initial identity from a pool of available identities. The identity is then utilized to perform subsequent activities and all such activities are logged against that identity. In a social network application, this identity is distributed to other members of the social group to allow continued interaction amongst the group. The assigned identity is frequently changed and redistributed to the group.

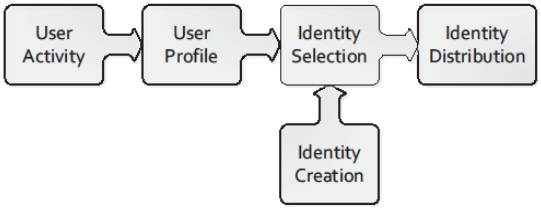


Figure 1. Identity hopping framework

2.1. Identity hopping techniques

Hopping has been used in communication to enhance security for decades. It is the core of the popular Frequency Hopping Spread Spectrum (FHSS) technique used in Wi-Fi and 3G mobile networks. Initial application of the frequency hopping technique dates back to the 1950s in which it was used to prevent 3rd parties from intercepting communications [4]. The sender and receiver would use a pseudo-random sequence of frequencies that are known to both of them. Unless the interceptor knows the sequence and timing, it would not be possible to intercept the communication [5].Similar hopping technique applied to identities instead of frequencies can be used to enhance identity security in social networks.

Multiple possibilities are available to perform identity hopping. The selection of an identity hopping method largely depends on the application. In general, all identity hopping techniques are designed to ensure that each identity does not contain sufficient activity to risk identity security and that identities cannot be correlated together to discover the full activities performed by the user.

With sufficiently large number of available identities, the users can utilize a pseudo-random sequence of identities and change to a new identity with every new activity or after a preset interval of time. Frequent hopping is key with this technique to ensure that activities in each of the identities are not sufficient for the identities to be correlated. The advantage of this technique is that the identities used do not need to be exchanged each time the identity is changed. The user needs only to distribute the initial identity, pseudo-random number seed, and the timing for hopping. The disadvantage of this technique is the need of a large number of identities to ensure the security of a user's identity.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3n3.41 *e-ISSN: 2251-7545*

```
Input: SR: Max Identity Security Risk, CR: Max Correlation Risk,
T: Max Time without Identity
1:  if !identityActivated() then
2:      currentIdentity = selectIdentity(currentProfile)
3:      activateIdentity(currentIdentity)
4:      broadcastIdentityToGroup(currentIdentity)
5:      addToSavedProfiles(currentProfile, currentIdentity)
6:  else if identitySecurityRisk(currentProfile, savedProfiles) > SR then
7:      deactivateIdentity(currentIdentity)
8:      startTimer(T)
9:      while correlationRisk(currentProfile, savedProfiles) > CR and
        !timerExpired() do
10:         sleep()
11:     end while
12:     currentIdentity = selectIdentity(currentProfile)
13:     activateIdentity(currentIdentity)
14:     broadcastIdentityToGroup(currentIdentity)
15:     addToSavedProfiles(currentProfile, currentIdentity)
16: else
17:     sleep(t)
18: end if
```

Algorithm 1. Algorithm for identity hopping

When the number of identities available is limited, we utilize a different method for identity hopping depicted by Algorithm 1. The user starts with an initial identity to perform activities. As the identity is used for more activities, the security risk increases. Before performing a new activity with the current identity, the framework would asses whether performing the new activity would increase the identity security risk beyond a predefined security risk limit. If the identity security risk is expected to increase beyond the limit, then the user is assigned a new identity.

The identity security risk parameter and limit are application dependent and cannot be generalized. This could be a simple measure or a combination of various other parameters. For location tracking applications, as an example, the identity security risk parameter can be based on distance. For online browsing, this could be based on a combination of several parameters such as the number of sites visited and site category (online store, news, etc.). The identity security risk limit at which the identity is changed should also be dynamically adjusted based on the user activity.

To guard against the ability to correlate the various identities together, a correlation risk parameter is also established. The parameter measures the relative ability to correlate together two profiles that encompass different user activity. Correlation can be easy in cases where a parameter of the profile is constantly changing in one direction and the identity is changed in the middle. The identities can be correlated based on the pattern of that parameter change. An example of such parameter is the movement on a map with predefined roads. Utilization of the correlation risk limit will introduce a gap of time during which activities are not logged against an identity. Resumption of identity utilization will only happen when the activities have changed significantly enough to reduce the risk of correlation.

The combination of the identity security risk limit and the correlation risk limits can be adjusted dynamically based on users' behavior and preferences. If the user exhibits a small range of activities, then the values of the parameter limits can be adjusted downwards to increase the number of identities being used. In cases where there are no more new identities to be utilized, the new activity can be added to the identity that would result in the least amount of identity security risk.

2.2. Identity creation

Creation of a set of identities manually is sufficient for this framework. The user can create the identities and provide the list to the framework for selection during hopping. For maximum privacy, it would be best to automatically create the identities on demand if allowed by the application. In this case, the privacy and correlation risk limits can be set to minimum values which would cause the framework to create and use a higher number of identities. In both cases, users have to use an IP changer like JonDo [6] software to prevent any possible correlation between their real identities and the newly created identities and to prevent correlation between the new identities themselves.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3n3.41 *e-ISSN: 2251-7545*

```
Input: profile: profile to be match, savedProfiles: list of all profiles
that have been used so far, SR:Max Identity Security Risk
Output: identity matching the provided profile
1:  procedure
2:      { check if we can use one of the existing identities : }
3:      for all savedProfiles do
4:          if  identitySecurityRisk(profile, savedProfile) < minSR  then
5:              minSR  = identitySecurityRisk(profile, savedProfile)
6:              selectedProfile = savedProfile
7:          end if
8:      end for
9:      if (minSR ≤ SR) or (no more identities left) then
10:         Return selectedProfile.identity
11:     else
12:         Return new identity
13:     end if
14: end procedure
```

Algorithm 2. The selectIdentity() function

2.2. Identity distribution

In the case where identity hoping is not based on a pseudo-random sequence, then the identity will need to be distributed to the members of the social network group. This distribution should be done through a medium that differs from the social network application to achieve maximum identity security. In our approach, instant messaging applications have been chosen as a distribution medium.

In order to protect the users' identities against unwanted disclosure, the identities may only be transported in an encrypted form using asymmetric key algorithms, e.g. RSA algorithm. To perform encryption and decryption, each user must have access to other users' public keys. To achieve the public keys exchange in a secure manner, a user needs to generate a "one way" public and private key to be used in the invitation process. Figure 2 shows the distribution of private and public keys for a group of users. Bob, John, and Alice form one group, while Alice and Iva form a different group. In addition to his own private key, Bob has the public keys for all members of the groups he participates in.

3. Case study

To illustrate the effectiveness of our proposed solution, we have implemented it for Latitude, a location-aware mobile application developed by Google [7]. Latitude allows mobile phone users to share their locations with certain people. Using a Google account, the user's cell phone location is mapped on Google maps. The user can control the accuracy and details of what each of the other users can see: an exact location can be allowed, or it can be limited to identifying the city only. Users may only see the location of those friends who have decided to share their location with them. Recently, Google Latitude has been enhanced to optionally record history of places visited and to accumulate time spent at each place. This information is then used to display statistics such as "Time at Work", "Time Spent at Home" and "Time Spent Out".

Figure 3 displays a snapshot of Google Latitude for a group of friends: Bob, Alice, and John taken from John's Android mobile phone. The map shows the location of Bob in Dusseldorf, the location of Alice in Leverkusen, and the location of John in Cologne.

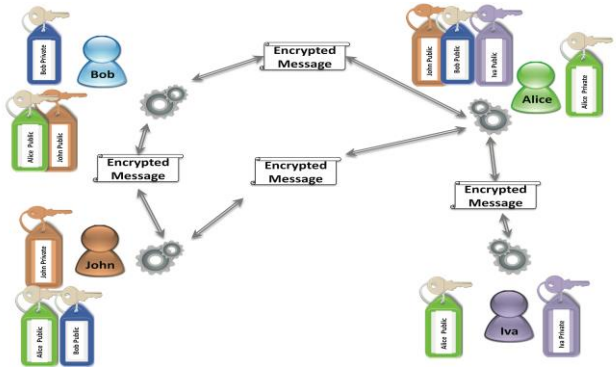


Figure 2. Distribution of private and public keys for a group of users

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3n3.41 *e-ISSN: 2251-7545*

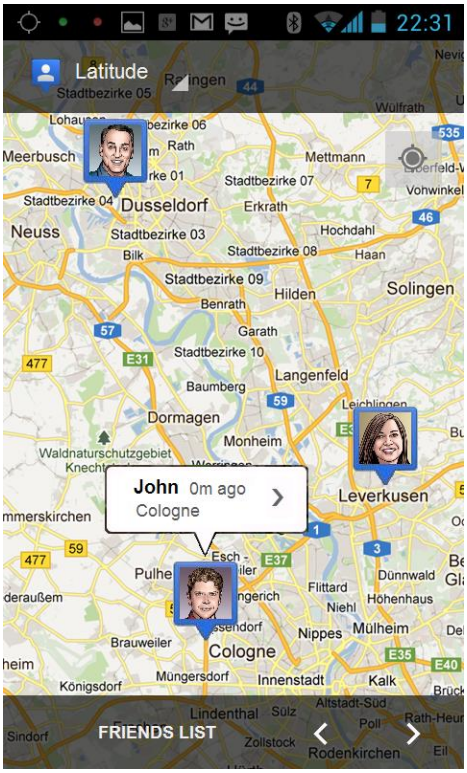


Figure 3. Distribution of private and public keys for a group of users

Users have to choose explicitly to join Latitude and permit having their current location displayed on Google maps and viewed by others. Several options are available in Latitude to address concerns of location privacy: Latitude can be turned off by the user, the location can be manually entered by the users to hide their current locations, additionally, Google announced that Latitude overwrites user's previous location with the new location data, and does not keep logs of locations provided to the service. Nevertheless, the provided options are not enough to resolve the main concern with Latitude. Users are releasing to Google their instant locations, traveling habits, and times spent at different places using their real identities. If the data gathered by Google is abused, it can potentially present not only a privacy risk but also a security risk to users [8].

To resolve the main issue with Google Latitude, we integrated our proposed identity management framework described in Section 2 using the Google Latitude API [9]. We have implemented the framework and tested it with real users. Similar to Latitude, the framework can be run on desktops, laptops, or on mobile devices.

Users are responsible for creating and managing identities themselves using functions in the framework. The IP Changer JonDo [6] is used to create the identities to prevent correlation between users and their identities. Users are also required to enter their home and work addresses and associate some identities with them. Additionally, to prevent possible correlation between the users and identities, the same identity is used for the user while staying at frequently visited places like home, work, etc.

The main algorithm of the framework consists of three main parts as shown in Algorithm 3. If the user is not signed in, an identity is selected for the user from the available list of identities using the method described in Section 2.1. Once an identity is selected, the user is signed in and his or her identity is broadcasted to the friends via a secure encrypted instant message as described in Section 2.3.

Once the distance traveled by a user is more than R, the user's identity has to be switched. To prevent any possible correlation between the old identity and the new identity, a guard band is used. The guard band consists of a timer and a distance. The timer is controlled by T and the guard band distance is controlled by G. All input parameters (R, G, and T) are controlled either by the user or his traveling habits.

```
Input: R, T, G
1:  if !signedIn() then
2:        selectIdentity(currentLocation)
3:        signIn()
4:        broadcastIdentityToGroup()
5:        saveCurrentLocation()
6:  else if distanceTraveled() > R then
7:        signOut()
8:        startTimer()
9:        saveCurrentLocation()
10:             while distanceTraveled() < G
and !timerExpired() do
11:             sleep(T)
12:       end while
13:       selectIdentity(currentLocation)
14:       signIn()
15:       broadcastIdentityToGroup()
16:       saveCurrentLocation()
17: else
18:       sleep(T)
19: end if
```

Algorithm 3. Algorithm for Google Latitude identity hopping

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.41 *e-ISSN: 2251-7545*

In this case study, the parameter R represents the identity security risk factor; larger R values mean that more activities are tied to a single identity, thus increasing the identity security risk. The correlation risk factor is represented by a combination of the parameters G and T; a larger value of G and T increases the separation between the identities which results in a lower ability for outsiders to correlate the various identities together reducing the correlation risk factor.

Figure 4 shows the preferences for the framework. The first part is for password management and disabling the location tracking service. The second part is for managing identities. Users can add, create, delete, and disable identities. Note that identities are created manually by users. Users need also to set up their home and work addresses and assign some identities with them. The application interface also allows users to create additional frequently visited places. The input parameters (R, G, and T) for the main algorithm used by the interface can be manually set in the preferences. If the parameters are not set, they will be automatically adjusted based on the traveling habits of the user.

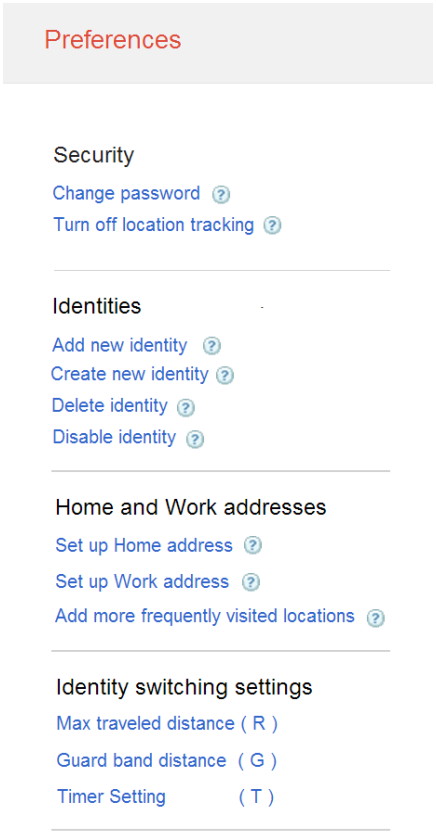
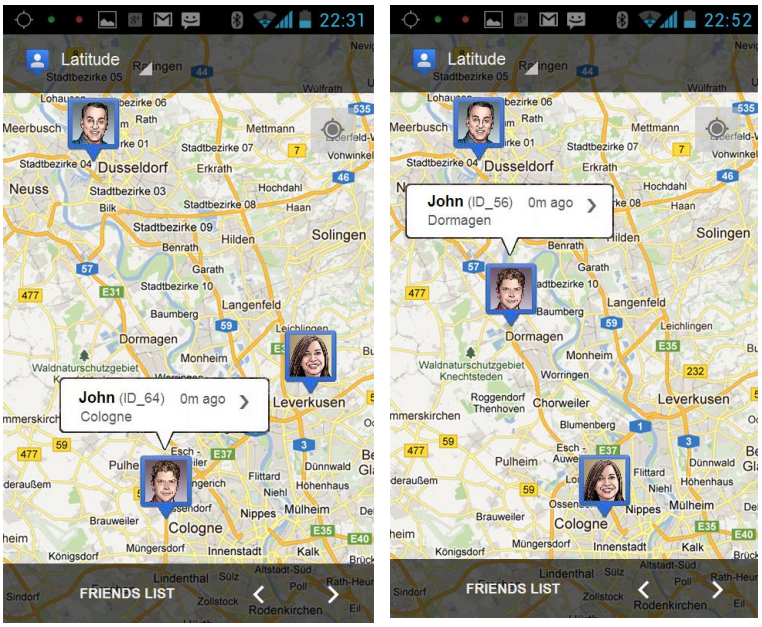


Figure 4. A snapshot of the preferences settings screen

Figure 5 shows snapshots of the framework for the same group of friends as in Figure 3: Bob, Alice, and John. The actual used identities are shown next to the users' names. The left graph, Figure 5(a) shows the initial locations for Bob, Alice, and John while the right graph, Figure 5(b) shows their locations twenty minutes later. Alice traveled from Leverkusen to Cologne and her identity has changed. John traveled from Cologne to Dormagen and his identity has also changed. Bob remained stationary at work and his identity remained the same.

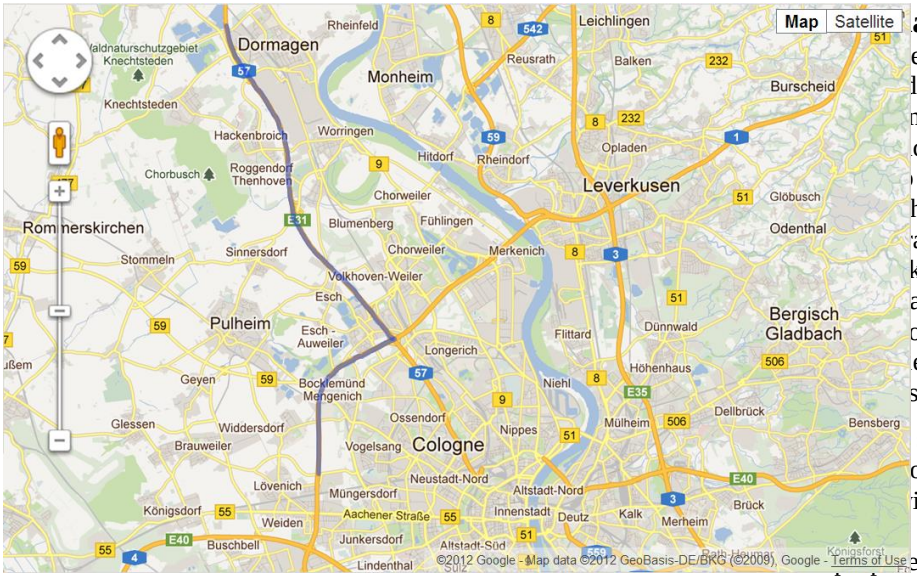


(a) (b)

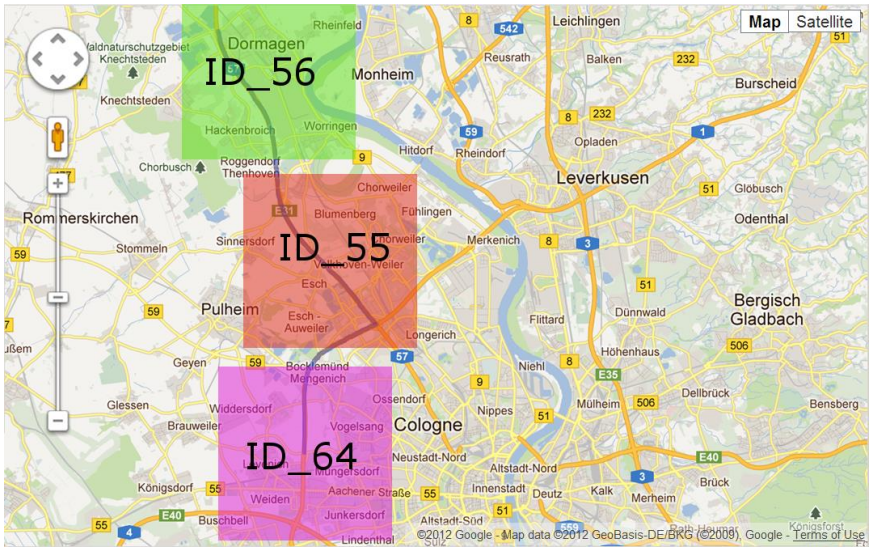
Figure 5. A Snapshots of the security framework for Google Latitude for a group of users as viewed on John's Android mobile

Figure 6 illustrates identity hoping by showing the different identities used for John during his travel from Cologne to Dormagen. The top graph, Figure 6(a), shows the route used in his trip. The bottom graph, Figure 6(b), shows the identities used for John during his travel. Each identity covers a square area and is used whenever John is traveling in that area.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3n3.41 *e-ISSN: 2251-7545*



(a) John's travel route from Cologne to Dormagen



(b)Areas for identities used during John's trip from Cologne to Dormagen

Figure 6. Areas for identities used during John's trip from Cologne to Dormagen

To evaluate the effectiveness of the framework in preserving privacy and reducing security risks, we use the technique proposed in [10] to measure the loss of privacy by the amount of new information that the system was able to gain about the person. Table 1 summarizes the various types of information that can be accessed in Google Latitude compared to our proposed framework.

ated Work

ecent years, many researchers have l on privacy and security for online networks due to their high practical ce. Generally, research work focused main streams. In the first stream, hers proposed designs of fully alized privacy aware online social ks that replace the traditional server- architectures. PeerSon system [11], ok [12], and Vis-a-Vis [13] are es of such peer-to-peer online social s

second stream focused on proposing ounds to enhance privacy while ing the server-based model and services. In [14], the authors ed NOYB, an encryption tool useful to hide the real users profile information from the data center. Lockr, a discretionary access control system, has been proposed in [15] to decouple the social information from the content the users share with others. In this approach, the list of friends is not explicitly stored anywhere; instead, every user distributes a signed social attestation to each of her social contacts; only users that have a proper social attestation are allowed to access the resources. Other approaches like FlyByNight [16] and Persona [17] were also proposed to mitigate identity security risks in social networks using encryption to conceal content to the provider. Those techniques can be combined with our proposed architectural framework for maximum privacy.

Some research has focused primarily on protecting users' location privacy. Beresford and Stajano developed [18], [19] the mix zone concept. In their model, they assume the existence of a trusted middleware system, positioned between the underlying location system and un-trusted third-party applications. Applications register interest in a geographic space with the middleware. Users register interest in a particular set of location-aware applications and the middleware limits the location information received by applications to location sightings of registered users located inside the application zone. Mokbel et al proposed Casper [20], a centralized location privacy-preserving framework in which mobile users can entertain location-based services without revealing their

Doi: 10.7321/jscse.v3.n3.41 *e-ISSN: 2251-7545*

location information. In their proposed framework, they use a location anonymizer and a privacy-aware query processor to generate cloaked spatial regions based on user privacy profile that cannot be used to give any information about the exact location of the mobile user. As opposed to their solution, our framework does not rely on a trusted third party that could become the system bottleneck.

Other research has focused on privacy enhancing identity management techniques to protect users' privacy in an electronic society [21], [22], [23]. Generally, most privacy enhancing techniques allow a user to control the nature and amount of personal information disclosed based on communication network providing anonymity. Our work is orthogonal to the proposed techniques and can be combined in our solution framework.

5. Conclusions

In this paper, we proposed an identity security framework for interfacing mobile device applications to social networks. The framework utilizes identity hopping techniques and anti-correlation measure to enhance identity security and privacy. We have implemented the proposed interface on the Google Latitude application as a case study. The implementation showed the effectiveness of the proposed interface in enhancing identity security and privacy.

References

[1] International Telecommunication Union, "Key Global Telecom Indicators for the World Telecommunication Service Sector", available online at: http://www.itu.int/ITUD/ict/statistics/at glance/KeyTelecom.html, 2011.

[2] E. Mikalajunaite, "Android Market Reaches Half a Million Successful Submissions", available online at: http://www.research2guidance.com/android-marketreaches-half-a-million-successful-submissions, 2011.

[3] J. Angwin and J. Valentino-Devries, "Google's iPhone Tracking", The Wall Street Journal, Feb 2012.

[4] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of Spread-Spectrum Communications - A Tutorial", IEEE Transactions on Communications, vol. Com-30, no. 5, pp. 855–884, May 1982.

[5] A. Goniotakis and A. K. Elhakeem, "Security Evaluation of a New Analog Speech Privacy/Scrambling Device Using Hopping Filters", IEEE Journal on Selected Areas in Communications, vol. 8, no. 5, pp. 781–799, 1990.

[6] JonDo, "JonDo - The IP Changer", available online at: https://anonymousproxy- servers.net/en/jondo.html, 2012.

[7] Google, "Latitude", available online at: http://www.google.com/latitude, 2012.

[8] T. Claburn, "Google Latitude Spurs Privacy Backlash", Tech. Rep., Feb. 2009.

[9] Google, "Google Latitude API Reference Guide", available online at: https://developers.google.com/latitude/v1/, 2012.

[10] L. Longpre and V. Kreinovich, "How to Measure Loss of Privacy", University of Texas at El Paso, Tech. Rep., 2006.

[11] S. Buchegger, D. Schioberg, L.-H. Vu, and A. Datta, "PeerSoN: P2P Social Networking: Early Experiences and Insights", in Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, Nuremberg, Germany, Mar. 2009, pp. 46–52.

[12] L. A. Cutillo, R. Molva, and T. Strufe, "Privacy Preserving Social Networking Through Decentralization", in Proceedings of the Sixth International Conference on Wireless On-Demand Network Systems and Services, Snowbird, Utah, USA, Feb 2009, pp. 133–140.

[13] A. Shakimov, A. Varshavsky, L. P. Cox, and R. Caceres, "Privacy, Cost, and Availability Tradeoffs in Decentralized OSNs", in Proceedings of the 2nd ACM Workshop on Online Social Networks, Barcelona, Spain, Aug. 2009, pp. 13–18.

[14] S. Guha, K. Tang, and P. Francis, "NOYB: Privacy in Online Social Networks", in Proceedings of the First Workshop on Online Social Networks, Seattle, WA, USA, Aug. 2008, pp. 49–54.

[15] A. Tootoonchian, K. K. Gollu, S. Saroiu, Y. Ganjali, and A. Wolman, "Lockr: Social Access Control for Web 2.0", in Proceedings of the First Workshop on Online Social Networks, Seattle, WA, USA, Aug. 2008, pp. 43–48.

[16] M. M. Lucas and N. Borisov, "FlyByNight: Mitigating the Privacy Risks of Social Networking", in Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society, Alexandria, Virginia, USA, Oct. 2008, pp. 1–8.

[17] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an Online Social Network with User-Defined Privacy", in Proceedings of the ACM

Doi: 10.7321/jscse.v3n3.41 *e-ISSN: 2251-7545*

SIGCOMM 2009 Conference on Data communication, Barcelona, Spain, Aug. 2009, pp. 135–146.

[18] A. R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing", IEEE Pervasive Computing, vol. 2, no. 1, pp. 46–55, Jan. 2003.

[19] A. Beresford and F. Stajano, "Mix Zones: User Privacy in Location-aware Services", in Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004., march 2004, pp. 127–131.

[20] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The New Casper: Query Processing for Location Services without Compromising Privacy", in Proceedings of the 32nd International Conference on Very Large Databases, ser. VLDB '06. VLDB Endowment, 2006, pp. 763–774.

[21] S. Clauß, D. Kesdogan, and T. Kölsch, "Privacy Enhancing Identity Management: Protection Against Re-identification and Profiling", in Proceedings of the 2005 Workshop on Digital Identity Management. New York, NY, USA: ACM, 2005, pp. 84–93.

[22] Prime, "Privacy and Identity Management for Europe", available online at: http://www.prime-project.eu/, 2012.

[23] PrimeLife, "Bringing Sustainable Privacy and Identity Management to Future Networks and Services", available online at: http://www. primelife.eu/, 2012.

* Corresponding Author:
Ahmad Zmily,
School of Information Technology and Engineering,
German Jordanian Unvierstiy, Amman, Jordan,
Email: ahmad.zmily@gju.edu.jo Tel:+962-64294121

Table 1. Information loss comparison between Google Latitude and the proposed security framework

Information	**Latitude**	**Framework**
Real Identity	Accessible	Hidden using identity hoping
Residential Address	Accessible	Linked to a single fake identity makes it hard to correlate with the real identity
Work Address	Accessible	Linked to a different fake identity makes it hard to correlate with the real identity or the home address
Visited Places	Accessible	Identity hoping prevents system from tracking places visited by a user
Habits (Shopping, Driving, Eating, etc)	Accessible	Different identities are used for different places make it hard for the system to correlate activities with a single user
Friends	Accessible	Real identities of friends are hidden