

Social Email: A Framework and Application for More Socially-Aware Communications

Thomas Tran, Jeff Rowe, and S. Felix Wu

Department of Computer Science
University of California, Davis
{`trantho, rowe, wu`}@`cs.ucdavis.edu`

Abstract. As online social networks (OSN) attempt to mimic real life social networks, we have made progress towards using OSNs to provide us with data to allow for richer and more robust online communications. In this paper, we present a novel approach towards socially-aware email. Currently, email provides too little control to the recipient. Our approach, dubbed SoEmail, provides **social context to messages** using an OSN's underlying social graph. This not only gives the recipient control over who can message her, but it also provides her with an understanding of where the message originated from, socially. Furthermore, users who do not wish to use the built-in social aspect of SoEmail, can send and receive emails without any changes to their behavior. The modifications we made to the email application to provide this social context are not invasive and can be easily ignored by any existing email client. We built SoEmail on top of an existing framework, known as Davis Social Links, which allows SoEmail to be completely agnostic about the underlying OSN. We created a web-based, standards-based web client for SoEmail using Facebook and Gmail as the underlying systems which has been released for public use and has had a good adoption rate.

1 Introduction

As the Web becomes more interactive, a number of venues exist to facilitate communication and meeting new people. The most ubiquitous of these medias are online social networks (OSNs) which has immensely grown in popularity. While OSNs do a great job of mapping out a user's social connections, the friendships are, in general, symmetric and binary. It is binary in the sense that two users either are friends, or they are not. There is no system for qualitatively rating friendships other than controls to allow specific friends to access parts of a user's profile. All friendships are essentially the same at the system level, which is obviously inconsistent with real world social networks. We believe that without an ability to rate friendship and trust, OSNs are incomplete. This is crucial in order to more accurately model how real world networking works. Our belief has been echoed by previous research by Krishnamurthy et al. [5] and Gross et al. [4].

Since the underlying graphs of OSNs is a rough reflection of real world social networks in a digital format, we believe that the small world hypothesis [8] also

should apply to OSNs. As a result, social networks can be used for more than just communicating with immediate friends. In fact, it is theoretically possible for a user to connect to any other user with about six intermediary friends. While it may be difficult to manually search for a set of connecting intermediary friends between two users, our system can efficiently and automatically find these social routes on behalf of the user.

In this paper, we attempt to rectify the current status of OSNs in order to improve online communication by adding social context. Our work in the past includes an instantiation of Davis Social Links[1] which adds a layer above OSNs (Facebook, in our implementation) that allow for extended features such as maintaining trust and finding social routes to other users. In this paper, we show how we improve upon DSL and develop a web-based, standards-based email client that adds in social context without being intrusive to the end users. In section 2, we discuss in more detail the problem statement and things that have been done in the past to alleviate some of the shortcomings of OSNs. In section 3, we present a quick overview of DSL and how it connects with SoEmail along a discussion on some improvements to DSL. Once the reader has an understanding of DSL, we then delve into how we implemented SoEmail in section 4. We continue the discussion in section 5 by examining challenges and points for improvement. Finally we conclude our paper in section 6 and highlight some avenues for future work related to SoEmail and DSL.

2 Problem and Prior Work

Social networks have gone a long way towards mapping social connections between friends and peers in the real world. However, they are currently inadequate at adding social context to user-to-user communication. Most social networks, such as Facebook [2] and MySpace [9], provides no social context for their messaging platforms at all. LinkedIn [7] has built in functionality to be introduced to another user through a mutual connection. However, we believe that even LinkedIn's feature is too conservative and narrow in its goals. Instead of only being able to communicate to users two hops away (i.e. a friend of a friend), we envision a system in which users can communicate to others through many intermediary hops. Furthermore, while social networks can model connections in real life networks, they are currently unable to practically rate friendships or model the level of trust between two users.

Finally, in most social networks, a user can control who communicates to them only by using very sweeping policies that are neither dynamic nor easy to control with great precision regarding who can communicate with the user and who cannot. For example, in Facebook, a user can specify who can message to them by setting a global policy regarding everyone who is not their friend and by blocking specific users.

To this end, we have previously designed and published work on a system which adds social context to multiple-hop messages named Davis Social Link (DSL). However, for reasons to be discussed in the next section, DSL was incomplete as a messaging platform. We will later see how SoEmail utilizes DSL's

functionalities to build a complete, more robust and standard-compatible messaging system.

3 Overview of Davis Social Links (DSL)

Since SoEmail relies on DSL, it would be useful to first give a brief overview of DSL along with mentioning some additional features since [14].

Davis Social Links (DSL) is built on top of existing OSNs. Users can sign up for DSL service, which provides their social graph to DSL, allowing the system to help the user establish social context in online communications. In our proof of concept, we have utilized Facebook's API to gather data on each user's social connections. Using this information, DSL publishes its own API through SOAP. This allows third party applications to leverage the power of social networks while being relatively agnostic about which network (eg. Facebook vs. MySpace) it is using. Furthermore, DSL differentiates itself from other APIs such as OpenSocial [10] by providing additional functionality that the other APIs do not. The biggest amongst these are the social router and the trust management system, which we will focus our overview on.

3.1 Trust Management

In today's OSNs, friendships are very basic: they are both binary and symmetric. They are binary in the sense that two people can either be friends or not. There are no levels of friendship, although many social networks have built in privacy controls that allow users to prevent certain friends from being able to access sensitive profile information or pictures. Unfortunately, this modeling of friendship is not very comparable to real world friendships. Therefore, DSL modifies each friendship, or edge, by adding a trust value τ between 0 and 1 in accordance to [12]. Trust, in our system is asymmetric, meaning that while a user, Alice, may completely trust her friend, Bob, Bob may not feel the same way about Alice. Trust is currently roughly the ratio of good messages (as rated by the message recipients) to total messages. When an edge's trust value is below a threshold, messages become dropped with probability equal to $1 - \tau$. This deters spammers as each piece of spam they send out drops their trust value with their friends, which will eventually prevent the spammer from messaging others as their messages will have a high chance of being dropped. The spammer may remedy this situation by sending out good messages in addition to spam, but the spammer must send about two good messages for each piece of spam, greatly hampering his spamming ability.

We show an example of this in Fig. 2. Here, we show values marking how much each person trusts the previous hop. For example, Bob completely trusts Alice and therefore Alice's trustworthiness, as judged by Bob, is 1.00. On the other hand, Bob is not very well trusted by Carol and his trustworthiness is only 0.40. Note that each person in the relationship may judge the other differently. In this instance, while Bob completely trusts Alice, Alice may not reciprocate. In fact, Bob's trustworthiness, as judged by Alice, may only be 0.30, for example.

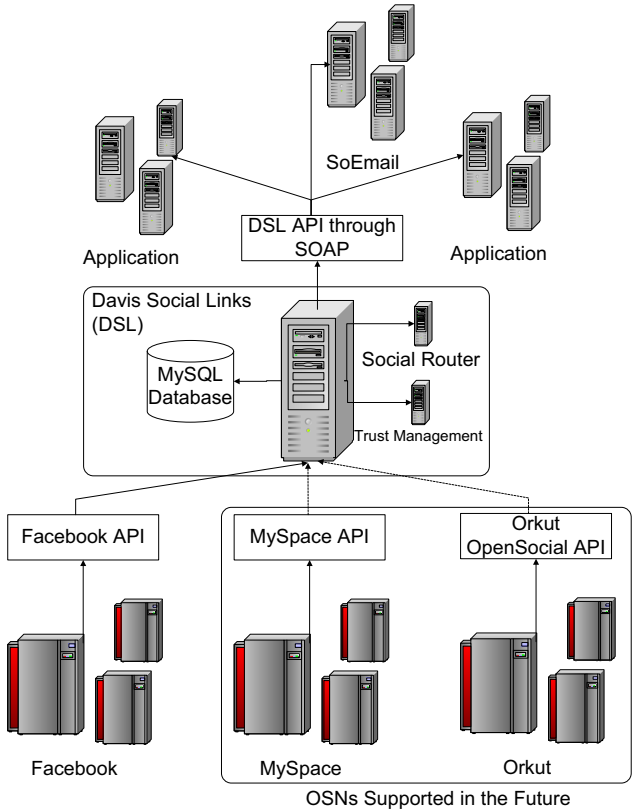


Fig. 1. Davis Social Links Architecture. Currently, DSL only supports Facebook, but it can be later expanded to handle different and even multiple OSNs. DSL is modular, and can be divided on to different servers to better handle high loads. DSL publishes an API through SOAP that other applications, such as SoEmail, can utilize to provide social context to the users.

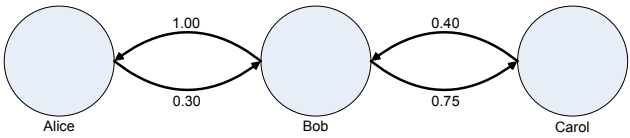


Fig. 2. An example social path. The arrows represent trust. For example, the arrow originating from Bob and going to Alice represent the trust that Bob places in Alice. Note that there are two arrows for each friendship, representing the fact that friendships are not symmetric. Here we see that two friends may have very different trust values for each other.

3.2 Social Router

In a social network, it is often advantageous to be able to communicate with people who are not direct friends. Facebook recognizes this and allows users to message people they are not friends with. However, these messages carry with them no social context and we argue that they carry no more weight nor should they be trusted more than an email from the same person. LinkedIn remedies this by allowing a mutual friend to introduce the message sender to the recipient, thus providing some social context to the messages that the recipient will be getting. We consider the recipient, in this example, to be two hops away from the sender while their mutual friend is one hop away from both. But what if a user wishes to communicate to someone more than two hops away? The social router in DSL will help the user by automatically finding a social path from the sender to the recipient, even if the recipient is an arbitrary number of hops away. Included in the social path are the trust values for each edge.

In addition to returning the social path, DSL also gives the trust value of each edge. As we noted in the previous subsection, if the trust between two nodes is sufficiently low, then the probability that the message gets through the edge connecting the two nodes is equal to τ . As a result, the probability that the message gets from its source to its destination is equal to the product of the probability that it gets through each hop. Note that it is NOT equivalent to the product of all the trust values, as if the trust value is high between two nodes, then the message will have 100% chance of reaching the next hop, even if the trust is not perfect (ie. the $\tau < 1$).

An application using DSL should then attach that social path to the message itself so that the recipient may understand where, socially, the message is coming from. Furthermore, the trust values will give the user a quick understanding of how credible the information in the message is. We will demonstrate how SoEmail utilizes the social path later on. The social router, coupled with the trust management system, forms the foundation of DSL and together, can effectively combat spam by excising the spammer by reducing his trustworthiness in relationship to his friends. This system is strong even against sybil attacks, as it is nontrivial to gain a new set of friends from scratch.

4 Social Email

4.1 Why Social Email?

In order to better understand how SoEmail can improve online communications, we examine a couple scenarios to compare SoEmail against standard email and communication through existing OSNs.

Receiving Message from Unfamiliar Sender. Through human networking, we often meet new contacts that may or may not be beneficial to our work or research. As a result, humans have created many social mechanisms to establish their credentials and reputation to a new person. In the scenario with Alice, Bob,

and Carol, where Alice and Carol do not know each other, Bob can provide one such mechanism. Bob can simply introduce Alice to Carol and by leveraging their trust of him, he can establish Alice's and Carol's credentials and reputation to each other. This is a very common social tool in the real world. In email, this can be emulated by having Bob email Carol, speaking on Alice's behalf. However, in the online universe, this can be greatly simplified and improved. Instead of having to find only one intermediary node for which both Alice and Carol are friends, we can find chains of nodes, linking Alice and Carol together. By leveraging the trust of that chain, we can ensure Alice and Carol's reputation to each other, without having to involve everyone in the middle. Furthermore, through SoEmail, we completely eliminate the need for the end users to know anything about the intermediary nodes. They can simply find a path, and send a message along it, with the understanding that SoEmail will modify the appropriate trust values automatically.

SoEmail vs. OSN Message Systems. The idea that we can automatically link complete strangers through a series of friends on an OSN is now new. LinkedIn already allows users to introduce themselves to a stranger through a mutual friend. However, unlike SoEmail, there is no feedback mechanism inherent in these OSN systems. If a user abuses the social graph to reach new users for the purposes of spam, there is little recourse. It may be difficult, for example, for the recipient of the spam to retaliate, especially if the sender is several hops away. However, SoEmail addresses this issue by making trust and reputation an integral part of the system, allowing the recipient to punish the sender and the sender's friends who are enabling the spam to be propagated.

4.2 Building SoEmail

At its core, SoEmail is a web-based, standards-based email client. We considered the usability of SoEmail and decided that the user should not have to learn anything about DSL or social routing if they did not want to. To this end, we strived to make SoEmail as similar to normal email clients as possible with the addition of the social context for users who are interested. SoEmail acts like any other email client: it allows users to access, send, and read emails in a manner consistent with their prior experience. Users do not have to learn anything new to use the basic functionality of SoEmail. We believe that other applications building on top of DSL can and should strive for the same goals. Since DSL does not intrude upon the communication itself nor upon the usage patterns, we believe that the addition of social context to any form of existing electronic communication (eg. Skype) should not sufficiently hinder the usability of the medium.

SoEmail is a web-based IMAP client built on top of SquirrelMail [13], a standards-based webmail package, which is also used by the UC Davis computer science department. We chose SquirrelMail because it was open source, simple and easy to customize. The web client connects to an IMAP server and can retrieve, read, and send emails like a normal client. We have further installed a plugin, multilogin [6], to allow SoEmail to support more than one



Fig. 3. The message sender can see the social path that will be used by SoEmail to send to the recipient along with the associated trust value. The shown path is a real one in our system between two DSL researchers.

email server. SquirrelMail, and as a result, SoEmail, is written in PHP, which is an excellent language for this project since it is easy to update and maintain and we can deploy SoEmail on any LAMP (Linux, Apache, MySQL and PHP) server, allowing for easy portability and replication. Since LAMP servers are so ubiquitous, SoEmail requires very little additional resources to run. SoEmail utilizes a MySQL database on the backend that stores the association between email address and Facebook ID along with, as we shall see, message IDs for each email sent through SoEmail to track the social path used by each email. As we will discuss later, we have leveraged the technology underlying SoEmail to allow for an easy port to the ProtoGENI clusters.

In addition to the plugin, our copy of SquirrelMail has been heavily modified to connect to DSL. Users can optionally use the social aspects of DSL in the client, adding social context to emails being sent out. SoEmail tags each message with an additional header field, X-DSL. This field contains a message identifier, which SoEmail associates with the path used by the sender. SoEmail tracks all the message identifiers and the corresponding social path used for that message. We decided against putting the social path in the header itself, since it can be easily forged or tampered. When the recipient opens the email in SoEmail, they can see the social context associated with the message. However, the user can open the email in any other web client without any compatibility concerns; the user would simply not see the social context as other web clients would not know what to do with the X-DSL field. As shown in Fig. 3, the message sender can see the social path that will be used by SoEmail to send to the recipient along with the associated trust value. We are currently exploring algorithms that would provide a few good social paths to the user so that they may further customize the social context of the message if they please. For example, there may exist a social path to the recipients along the sender's personal friends and another path along the sender's professional contacts. If the sender can choose which path to take, she can improve the quality of the social context being sent to the recipient.

To:

Cc:

Bcc:

Subject:

Priority: Receipt: ☐ On Read ☐ On Delivery

This is an example email.

Attach:

Fig. 4. The sender fills out the list of recipients that the email is intended for, he needs to only click one button, the "Find Social Paths" button to view the social path. Even if he were to not look at the social path, one will be found automatically and attached to the message. As a result, the sender can benefit from the social context provided by SoEmail and DSL without even realizing it.

4.3 SoEmail as a Service

SoEmail has been deployed as a service for anyone to use. As such, we strove to make it more robust than a typical proof-of-concept. We chose to support the Gmail servers first since they are by far the most popular email provider amongst our peers to have free IMAP servers. We are also working with the UC Davis computer science IT staff to add the UC Davis computer science department's server as well. We further encourage anyone interested in using SoEmail with a different email provider to contact us. The process for adding more email hosts has been streamlined and new ones can be added with relative ease.

In addition to ensuring that SoEmail caters to a large number of users, we also kept the DSL aspect of sending and receiving emails as un-intrusive as possible. As seen in Fig. 4, after the sender fills out the list of recipients that the email is intended for, he needs to only click one button to view the social path. Even if he were to not look at the social path, one will be found automatically and

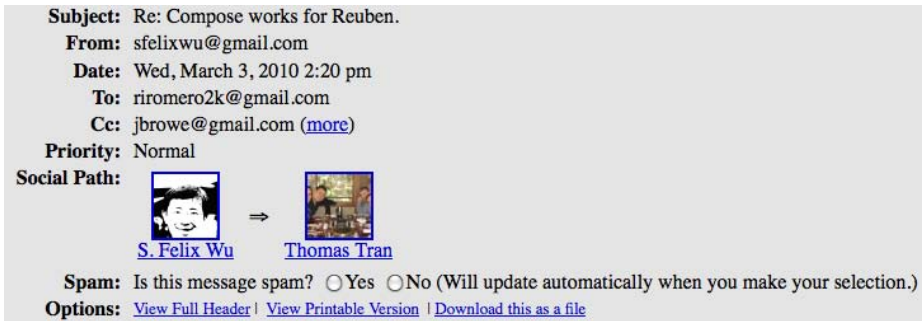


Fig. 5. A screenshot of the recipient's view of the message. Here, the social path is automatically displayed in the header, complete with links to the Facebook profiles of the nodes leading up to the recipient. This aids the recipient but for users who do not wish to utilize the social aspects of SoEmail can simply ignore the header and use the web application like any other email client.

attached to the message. As a result, the sender can benefit from the social context provided by SoEmail and DSL without even realizing it. In Fig. 5, we show a screenshot of the recipient's view of the message. Here, the social path is automatically displayed in the header, complete with links to the Facebook profiles of the nodes leading up to the recipient. This aids the recipient but for users who do not wish to utilize the social aspects of SoEmail can simply ignore the header and use the web application like any other email client.

In order to improve the functionality and usability of SoEmail, we strive to gather usage data and feedback from our users. To this end, we have created a Facebook fan page to gather feedback and functionality requests along with demonstrating SoEmail at various GECs and talking with people willing to try out our application at the conferences. Users have provided invaluable feedback regarding the system and their usage patterns have helped us identify areas where we can improve usability and stability. For example, many users have expressed concerns with providing their email password to a third party. While SoEmail never stores passwords, we understand their concerns and are working to hand over the system to the UC Davis computer science support staff, adding credibility and trust to the system since it will be managed by professionals who also handle the computer science email system.

While the application is still in its infancy, we have had moderate success getting users to test and utilize the system. Currently, there are 48 users with 55 email addresses associated with SoEmail, despite the fact that Gmail is the only IMAP server currently supported. These users have sent 277 emails to 574 different recipients.

SoEmail is currently open to anyone with a Gmail address and we invite anyone interested to join and test our system at <http://soemail.cs.ucdavis.edu>.

4.4 SoEmail on ProtoGENI

Our larger vision is to investigate techniques for incorporating application level social context into next-generation Internet architectures. To realize this, a framework for testing new social routing schemes that use information from social-network based applications is needed. We choose the emerging GENI [3] testing infrastructure as a basis for our investigations. In particular, we have installed and tested both our DSL trust based social routing service, and our SoEmail social email client in the evolving protoGENI [11] testbed under development at the University of Utah. ProtoGENI is an NSF-funded set of connected computer nodes users can utilize to experiment with new internet protocols. The protoGENI testbed provides an emulated testing environment similar to Emulab, but distributed across a variety of sites connected via high-speed backbone network links. To install our DSL service and SoEmail application, a small experimental testbed with several emulated hosts was created. One host was configured with the Red Hat 9 Enterprise operating system running an Apache 2 web server and a MySQL database to provide the DSL core service. We transfer our operational DSL database to this server running inside the protoGENI testbed and operate it as a mirror DSL trust base routing system. A second protoGENI host was configured as a SoEmail IMAP client and linked to the DSL server inside the testbed to provide the necessary social services. Finally, this second machine was configured to make the SoEmail client running inside protoGENI available to users in the current Internet. By distributing the DSL core service across multiple protoGENI sites, we can test new, next-generation Internet routing schemes using realistic networking components, and with traffic generated by external users interacting with SoEmail and other social networking applications from the existing Internet. Our experience shows no significant performance penalties from running inside protoGENI versus running on the Internet itself.

5 Discussion

While SoEmail has been stable so far, we recognize that there are some issues that we would like to address in the near future. Most of these issues have been brought up to us by our users. Below, we have them outlined roughly in ascending order of complexity.

5.1 Supported Email Providers

Many people have requested that we expand our list of support email providers. To this end, we have made efforts to streamline the process of adding new email servers to SoEmail and we are working closely with the UC Davis computer science IT staff to add our department's email. This has been an eye opening experience for us as some email servers, such as the UC Davis computer science one, requires additional protective measures not fully supported by SquirrelMail,

such as tunneling and VPN. However, our IT support staff has been extremely helpful with workarounds to these privacy methods. We expect to fully support the UCD CS email server soon.

Afterwards, we will apply our experience from setting up SoEmail to work with the UCD CS server to connect to other IMAP servers.

5.2 Preventing Replay Attacks through X-DSL Field

Currently, we cannot prevent someone from replaying the message with the X-DSL field or forging the field, setting the identifier to a different message that has been sent in the past. We would like to eventually replace the X-DSL field with a cryptographically secured string and token that cannot be forged, perhaps encrypting a timestamp in the field to ensure that the message was not forged. Some methods for this have been discussed in theory and we hope to settle on one and implement it by the publication date of this paper.

5.3 Slow Initial Load Time

Many users of SoEmail, ourselves included, have noticed that loading the initial page that lists all of our emails in our inbox takes an unusually long time. Rough experiments have put the time from the moment the user logs in to when the inbox finishes loading at around 30 seconds, much longer than what today's internet users are familiar with. This has had a very strong impact on the usability and user experience regarding SoEmail. While working with the UCD CS IT staff, they have confirmed with us that the issue also appears in their instantiation of SquirrelMail. Our goal is to contribute to the SquirrelMail project if possible by addressing and attempting to optimize the load times. This would not only be beneficial to SoEmail, but to all administrators who deploy SquirrelMail, including our IT staff here at UC Davis.

5.4 Trusting SoEmail with Passwords

Many users have expressed concerns with providing their email password to a third party. While SoEmail never stores passwords, as security minded researchers, we understand their concerns and are working to hand over the system to the UC Davis computer science support staff, adding credibility and trust to the system since it will be managed by professionals who also handle the computer science email system.

5.5 Social Path Privacy Concerns

When a user is selected as a node for a social path, some information about the user's friends (namely, that they are friends with the node preceding and proceeding them) and the trust values are revealed to the sender and recipient. Currently, we display all this information, but we have been discussing hiding the social path or limiting its visibility so that the sender and recipient can only see the nodes on the social path that are close to them.

5.6 Supporting More than One OSN

A few users believe that by adding in one or more social networks in addition to Facebook can really flesh out the social graph along with providing better, more meaningful social paths that possibly take advantage of both OSNs. This issue is also partially related to DSL. However, the solution for this would be non-trivial as each user must also link all of their profiles on different OSNs together. However, if we can achieve this goal, we believe that the data and paths we can provide to the users will be much richer.

5.7 Trust and What It Means

One point of confusion for users of SoEmail and any DSL related application is the meaning of the trust value. The numbers are somewhat relative and do not easily translate to real world understanding of trust. For example, what does a trust value of 0.75 mean to the user? More often than not, it just means that the trust is high, but no more than that. Most users We have been experimenting with associating definitions with ranges of trust (eg. a trust rating of 0.9-1.0 can be called "Extremely Strong Trust") as a way of aiding users to understand trust and its impact on the social context provided to them.

6 Conclusion and Future Work

As seen in Section 5, SoEmail has some room for improvement, but as demonstrated here in this paper and through user feedback, our application has been performing well and is well received by its users. Simulations have shown that our system should be stable with even up to 40% of users being spammers, giving the system incredible robustness against spammers. We built SoEmail with usability in mind, and while we have been asked for clarification on how things such as trust and social paths work, we have not had one user tell us that the interface is too confusing.

We would like to keep user satisfaction high by addressing some of the shortcomings of SoEmail that we have already discussed. In particular, we are confident that two of the more popular issues will be fully addressed soon and we are internally discussing solutions to the other issues. The continued improvement of SoEmail is important to us because not only is it used as a service by dozens of people on a daily basis, but also because we would like to use SoEmail as a means to gather more data on how trust changes over time in a real environment with real users. To that end, we also plan on adding more anonymizing, data collecting functionality within both SoEmail and DSL. Finally, we hope to recruit more users to the system so that we can get more data and feedback. We have been actively advertising SoEmail at the conferences we attend and will continue to do so through demonstrations or presentations.

Once we have SoEmail in a state that we are satisfied with, we would like to polish SoEmail to the point where we can hand it off to the UC Davis computer science support staff to deploy and maintain. This will lend credibility to the

system which would hopefully attract people otherwise deterred from passing SoEmail their password.

On the theory side of DSL, we have been working closely with the department of sociology to better map out online social networks to real life social networks. We would like to get a better understanding of social interactions and feedback in the real world so that we may model it better in DSL.

Acknowledgments

We would like to thank Ken Gribble for his tireless help with setting up the SoEmail server, configuring it to our specifications, and providing patient support as we break everything he set up. This research is funded in part by National Science Foundation under CNS-0832202, Intel, and an MURI grant from Army Research Office under the ARSENAL project and we would like to thank them for continued support.

References

1. Banks, L., Bhattacharyya, P., Wu, S.F.: Davis social links: A social network based approach to future internet routing. In: FIST 2009: The Workshop on Trust and Security in the Future Internet (July 2009)
2. Facebook, <http://www.facebook.com/>
3. GENI, <http://www.geni.net/>
4. Gross, R., Acquisti, A., Heinz, J.H.: Information revelation and privacy in on-line social networks. In: WPES 2005: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, pp. 71–80 (2005)
5. Krishnamurthy, B., Wills, C.E.: Characterizing privacy in online social networks. In: WOSP 2008: Proceedings of the 1st Workshop on Online Social Networks, pp. 37–42 (2008)
6. Lesniewski, P.: http://squirrelmail.org/plugin_view.php?id=148
7. LinkedIn, <http://www.linkedin.com/>
8. Milgram, S.: The small world problem. *Psychology Today* 61, 60–67 (1967)
9. MySpace, <http://www.myspace.com/>
10. OpenSocial, <http://www.opensocial.org/>
11. ProtoGENI, <http://www.protojeni.net>
12. Spear, M., Lu, X., Matloff, N., Wu, S.F.: Karmanet: Leveraging trusted social paths to create judicious forwarders. In: IFCIN 2009: Proceedings of the First International Conference on Future Information Networks (2009)
13. SquirrelMail, <http://squirrelmail.org>
14. Tran, T., Chan, K., Ye, S., Bhattacharyya, P., Garg, A., Lu, X., Wu, S.F.: Design and implementation of davis social links osn kernel. In: SNAS 2009: Proceedings of the Workshop on Social Networks, Applications, and Systems, held in conjunction with the 2009 International Conference on Wireless Algorithms, Systems and Applications (2009)