SBone: Personal Device Sharing Using Social Networks*

Pravin Shankar, Badri Nath, Liviu Iftode, Vancheswaran Ananthanarayanan, Lu Han
Department of Computer Science, Rutgers University
{*spravin, badri, iftode, vanchi, luhan*}*@cs.rutgers.edu*

Abstract

People own a number of personal devices whose state and resources might be of interest to others. Owners of these devices may be willing to selectively share them with their family, friends, and colleagues, but there exists no easy and secure mechanism to do so. In this paper, we propose SBone, an architecture that allows personal devices to share their resources and state with each other, seamlessly and securely, using a social network for authentication, naming, discovery and access control. The sharing mechanism is derived from the inter-personal relationships that exists among owners by virtue of their presence in online social networks. We present a case study of how SBone can be used to share *internet connectivity* resource between devices.

1 Introduction

Mobile phones are one of the most popular personal devices with over four billion worldwide [23]. Other personal devices include iPods, Wii game controllers, in-car PCs, GPS devices, tablet PCs (iPads) and TiVo players. These devices feature powerful processors and internet connectivity via 3G and WiFi interfaces, making mobile personal devices an increasingly important computing platform.

People tend to share their belongings with their friends. For example, people often share books and movies that they own with their close ones. Carpooling is a common way in which people share their cars with their acquaintances. With the growth of Online Social Networks (OSNs), people have started sharing personal data, such as songs, pictures and videos, with their virtual friends using sites like Flickr and Youtube. OSNs like Facebook, Twitter and Google Buzz also allow users to share their whereabouts, the news they learn, and the changes that occur in their lives.

This phenomenon of sharing can be extended to personal devices. On these devices, owners can store personal data, such as songs, pictures and videos. These devices are also capable of sensing environmental information, such as traffic and temperature, which might be useful to provide participatory sensing [7]. The state of these devices, such as ring status (vibrate, silent or normal), can help friends to decide when to call, to text or to send mail. In addition to stored data and state, personal devices also have resources such as internet connectivity, GPS, camera, and talktime, which, today, are available only to their owners. However, owners may want to share these resources with their friends. For example, the owner of a phone with internet access may want to share her internet connection with phones owned by her friends who are in the vicinity. Unfortunately, today, we are still far from a world where personal device sharing is truly seamless. Ubiquitous device sharing among people requires support for authentication, access control, device naming and discovery.

In this paper, we introduce SBone, an architecture that facilitates sharing of resources offered by personal devices among people who connected in a social network. SBone establishes trust between devices by leveraging the relationships that already exist between their owners in an online social network.

We posit that inter-personal trust relationships should become a backbone for personal device sharing. By linking devices with users who own and carry them, it is possible to provide better usability, security as well as trust. Online Social Networks provide a natural way to connect personal devices of friends. An increasingly vast majority of people who own personal devices are connected to each other by Online Social Networks. Facebook reports more than 400 million active users, of which more than 100 million currently access Facebook through their mobile devices [12]. Online social networks have become

*Rutgers University, Department of Computer Science, Technical Report DCS-TR-666, February 2010.

an extremely popular way for people to maintain online identities and trust relationships in the form of friend links. These social links between users are an alluring asset, which can potentially be leveraged to provide authentication, naming, discovery and access control between the devices they carry.

Using SBone, users can add their devices to the system in a secure manner, and specify access control policies for their friends to share the state and resources of these devices. Devices are assigned personal names, and users can query the system for devices based on attributes and relationships. Providing these features in a scalable and secure way creates new challenges in authentication, access control, naming and discovery of devices. We discuss these challenges in this paper.

We also present a case study describing how SBone can be used to enable internet sharing between mobile personal devices. Devices such as phones, laptops, PDAs and wireless access points typically have internet connectivity, by means of WiFi or 3G interfaces. On the other hand, devices such as phones (without 3G subscription), IPods, TiVo players, typically do not possess internet connectivity. In our case study, SBone allows devices with internet resource to share it with other devices in the social network. Once a device is connected, other devices can find it by querying their social network.

The rest of the paper is organized as follows: section 2 presents the related work. In Section 3, we introduce the SBone architecture. Section 4 presents SBone design in detail, and Section 5 describe the case study of internet resource sharing between devices using SBone. We discuss open issues and scope for future work in Section 6, and conclude in Section 7.

2 Related Work

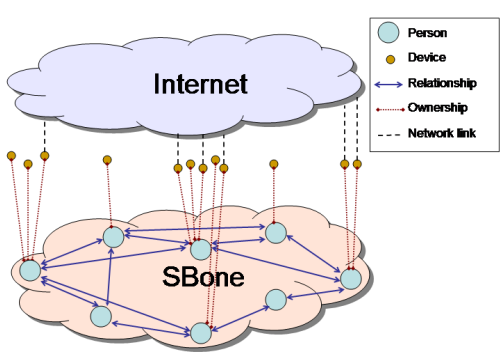
• **Device Communication.** In the realm of personal devices, the usability issues in home networking have been often highlighted, such as by Shehan and Edwards [29]. Several groups have proposed ways to improve the usability of device communication. Allman [1] introduced the idea of Personal Namespaces as an alternate and more usable and intuitive way to perform device naming. Allman, et. al. [2] introduced the idea of Relationship-Oriented Networking. MIT's Unmanaged Internet Architecture [13, 14] proposed a user-friendly architecture for naming, locating and routing between personal devices. Banks, et. al. [6] propose a routing scheme that involves social networks between nodes on the internet. The goal of all these works is to enable ubiquitous communication between devices. Our goal is to enable device sharing.

• **Device Sharing.** Several models for remote device access have been proposed [16, 32, 18, 19]. Hirofuchi, et. al. [16] proposed USB/IP, a peripheral bus extension over a TCP/IP network to access a remote USB device in a homogeneous environment. Kwon, etl al. [19] extended the architecture to support heterogeneous environments. Kong, et. al. [18] designed CameraCast, a system that enables access to a remote video sensor device. Wu, et. al. [32] proposed ComposableIO, a solution that enables applications to share IO peripherals over a network cloud. All these systems focus on remote access to peripheral IO devices. Peek and Flinn **??** propose Ensemblue, a distributed file system for personal devices that store multimedia. To the best of our knowledge, ours is the first system that enables device sharing using relationships between users in online social networks, and encompasses all aspects of communication including authentication, naming, discovery and access control.

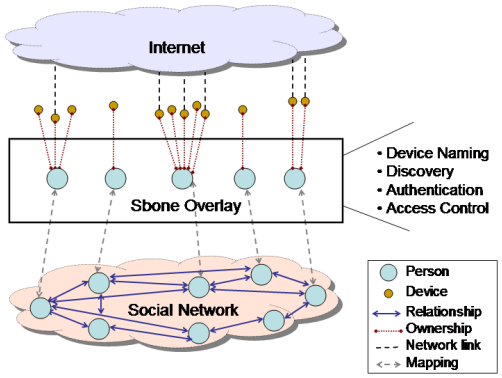
• **Device Authentication.** Balfanz et. al. [5] were the first to introduce the idea of using a location-limited channel such as an infrared link, USB key, gestures (touching the two devices together), etc. to authenticate ad-hoc devices. Balfanz et. al. [4] further introduced the idea of performing user-friendly wireless authentication by making use of location-limited channels. Asokan, et. al. [3] propose a way to manage access control for visitors to a personal wireless network. Dourish, et. al. [10] highlighted the importance of usability in designing secure systems. Wi-Fi Protected Setup (WPS) [31] is a standard that was established by the Wi-Fi Alliance for easy and secure establishment of a wireless network, by automatically setting up keys.

• **Authentication using Social Networks.** Social networking sites are commonly used by applications to bootstrap their own authenticated communication channels. Ramachandran and Feamster [25] proposed such a framework that allows applications to authenticate and discover peers using social networking sites. Majority of the large social networks today support such a functionality. Google's OpenSocial [22] and Facebook Connect [11] are application programming interfaces (APIs) for web-based social network applications that aim to make applications social networking enabled by letting users bring their identity and connections everywhere. OpenID [22] is an open, decentralized standard for authenticating users which can be used for access control, allowing users to log on to different services with the same digital identity where these services trust the authentication body. So far, social networks have only been proposed as an out-of-band authentication and peer discovery channel for users and applications.

Several recent works [30, 24] propose using a social network as a encrypted data store, and storing user keys in the social network. While there are similaries between these works and ours, these works address a different goal, which is to protect user privacy when users share



(a) **SBone model: A social graph of users extended with their personal devices. Some devices are connected to the internet.**



(b) **SBone architecture: SBone is an overlay network of users and their personal devices, SBone maps to a social network of users, to infer inter-personal relationships among them.**

Figure 1: **The SBone model and system architecture.**

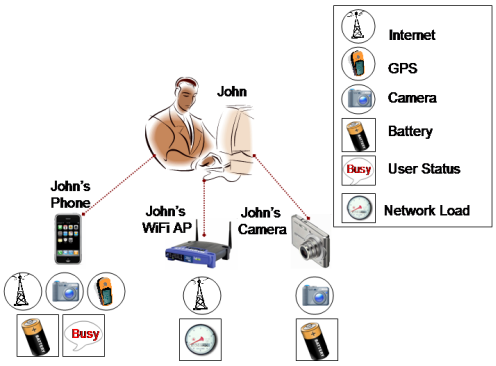


Figure 2: **A magnified view of an SBone user with the devices he owns. Each device has a set of resources (circles) and state information (squares).**

state in a social network. Our goal is to enable device sharing by making use of a social network.

3 SBone Architecture

Figure 1 describes the SBone model and system architecture. The SBone model (Figure 1(a)) consists of personal devices and users who own them. Edges between users represents a social relationship between them. Devices are connected to their owners by means of ownership links. Finally, some devices are connected to the internet, which is seen as a resource they can share with other devices.

The SBone architecture realizes this model by creating an overlay network of users and devices on top of an Online Social Network (OSN), as shown in Figure 1(b). In the rest of the paper, we refer to this overlay network, which connects to an online social network, and performs device naming, discovery, authentication and access control, as the SBone server.

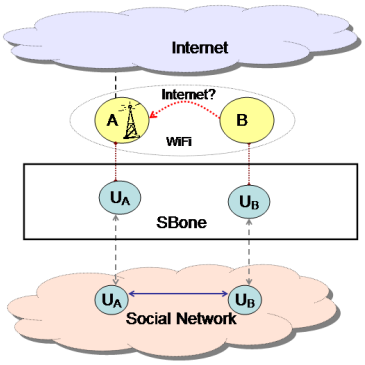
Each device has a set of resources and state information, as shown in Figure 2. Example of resources that devices may share are internet connectivity, GPS, camera, and talktime. State of a device could be either device-specific (such as battery, ring status) or environment-specific (such as traffic, temperature). This state is useful to other devices, and can be shared.

The SBone device sharing lifecycle consists of three phases: registration, connection and resource/state sharing. Registration establishes the ownership link for a new device that is to be added to SBone. Connection phase occurs whenever an SBone device tries to connect to the internet. Finally, a connected device enters the sharing phase, whenever it is available.

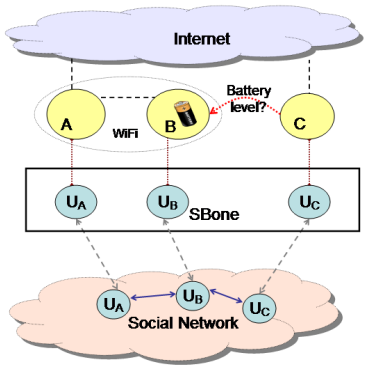
Registration: Registering a new device to SBone links it to its owner and assigns it a persistent personal name. The user then sets access control policy for this device's resources and state.

Connection: Devices look continuously to connect to the internet using either their own network link, or a friend's device that is willing to share its internet resource. On finding the latter, the two devices securely pair with each other using SBone. Once a device connects to the internet, it gets a routable name, which can be used by other devices to reach it.

Sharing: Personal devices discover the resources offered by other devices in SBone, using queries based on relationship and attributes. Once devices find each other, secure device-to-device pairing occurs. Unlike the traditional definition of pairing, device-to-device pairing in SBone refers not only to near-field communication, but



(a) **Resource sharing in SBone: Owners of devices A and B are friends. Device A has a sharable internet resource. SBone enables device B to get connected to the internet using A's connectivity.**



(b) **State sharing in SBone: Owners of devices B and C are friends. Device B is connected to the internet using A's connectivity. Device C wants to check B's battery level before making a call. SBone enables device C to query the state of device B.**

Figure 3: **Resource and state sharing in SBone.**

also to the case when the devices communicate via internet.

Resource and state sharing in SBone is illustrated in Figure 3(a) and Figure 3(b), respectively. In the example shown in Figure 3(a), devices A has a sharable internet resource. Devices A and B allow resource sharing between each other, since their respective owners are connected to each other by a social relationship. SBone enables device B to gets connected to the internet using device A's internet resource.

Devices can also share state information with each other, as illustrated in Figure 3(b). Device C wants to know , the battery level of device B, to decide whether to call, text or send mail. Since U_B and U_C are friends, the two devices are paired, and SBone enables device C to check B's battery state.

4 SBone Design

We now describe the design of SBone, focusing on the challenges faced in each step of the SBone device sharing lifecycle described in Section 3, and how SBone addresses these challenges.

4.1 Registering a Device

The first step in SBone device sharing is to introduce a new device to the SBone network, by registering it to one or more social networks that the user belongs to. The system must verify that the user owns the device. Systems that support mobile clients typically make use of a trusted out-of-band authentication channel, such as SMS, to allow the user to prove that she possesses the device. However, not all devices have the ability to send and receive SMSes.

Our goal is to support a broad variety of personal devices, and not limit the system to mobile phones. Therefore, we make use of location-limited channels[5], such as wireless, bluetooth, infrared or USB, to authenticate a device. For short-range communication, the social network cannot be used as an end point, so SBone allows the user to delegate her desktop, laptop or smartphone as an authentication proxy.

Once the device is registered, it generates a key pair and a public key certificate, which is stored in the newly added device. The device's public key is also stored on the SBone server, which acts as a certificate authority (CA).

Since a user may remove a device from the system (say when the device is lost, or the user no longer wants to use it), SBone has to maintain a certificate revocation list (CRL). A limitation of using CRL is that devices need to have connectivity with the SBone server at all times.

Devices owned by multiple users create a further challenge for the system. It is possible that multiple members of a family share a device. For simplicity, we assume that a personal device cannot have multiple ownership links.

4.2 Specifying Access Control Policy

Once a personal device is registered to the system, the user can specify which friends' devices can access what resource and state of this device. Specifying access control policy involves a trade-off between flexibility and usability. On one hand, an all-or-nothing policy can be adopted. The problem with such an approach is that, people do not have the same level of trust with all their

virtual friends in a social network. A friend in a social network can refer to an acquaintance, a co-worker, a close friend, sibling or spouse. On the other hand, a fine-grained access control is hard to specify for an average user. Typical users in a social network have several hundreds of friends, and it has been shown [28] that users typically do a bad job when entrusted with complex security decisions.

We approach this problem by representing the social network as a hypergraph, where each edge represents a *relationship*, such as friend, close friend, family member, sibling, co-worker, etc. Access control policy in SBone is specified per user, in the form of a matrix, where the rows are the resources and state vectors, and the columns are the relationships.

A typical user only has a limited number of relationship edges that she is a part of. This way, the number of rules that need to be specified for each device can be limited to a manageable number. Relationships can inherit other relationships. For example, the *close friend* relationship inherits the *friend* relationship, and the *sibling* relationship inherits the *family member* relationship. Users can define a precedence order among relationships, which is used between users who are connected to each other by multiple relationships.

There are two steps in the policy specification process: first, forming the relationship hypergraph, and second, specifying access control rules. The first step is performed each time a new friendship link is created in the social network. The second step is performed whenever a user adds a new device.

To form the relationship hypergraph, the system needs to understand the semantic meaning of an edge between users in a social network. This can be done by a combination of automated and manual techniques. First, the system mines the interactions in the social network to infer inter-personal relationships. Next, the system presents the inferred relationship hypergraph to users, and users can correct mistakes made by the system.

4.3 Naming and Addressing

SBone identifies each device using a personal name, that is specified by the owner, when registering the device. This name is unique within the device namespace of the user. Sbone uses the combination of user name and the personal device name to arrive at a globally unique personal name for every device. Figure 2 illustrates the device namespace of user John.

The device's personal name is persistent, and is used to identify the device, but it cannot be used to initiate communication to the device. Each time the device is connected, SBone needs to map the personal name to a routable address. The SBone client running on the device connects to the SBone server, and specifies this mapping.

If the device is connected to the Internet directly, this mapping can be done trivially. However, most devices are part of a home or an enterprise network behind a Network Address Translator (NAT), which makes it difficult for other devices to initiate connect to these devices. Similar challenge is faced by VOIP devices inside home and enterprise networks. We can use a STUN/TURN [27, 26] server to address this problem. STUN helps to discover the external IP address and port, which can be used to allow direct incoming connections. TURN is a relay server which relays messages if we are behind an Enterprise SNAT.

4.4 Discovering Devices

Users can access the device namespace of their friends, and browse their devices. They can also see what resources each device has, and its state vector, subject to the access control policy. In addition to manual browsing, a query interface is needed for two reasons. Firstly, the number of friends a typical user has is on the order of hundreds. Secondly, devices may want to directly discover each other without human intervention.

SBone supports a query interface to find other devices. A query can be formed using different attributes, such as, *find devices that have internet resource, and are within five miles of me*. Queries can also be based on relationship, such as, *find devices owned by my school friends*.

Relationships are also used for ranking query results. This is because a relationship is a measure of trust that the user has. Given the choice, a user would prefer sharing the resource of a device that belongs to a person she trusts more. Even if user does not specify relationships explicitly, SBone ranks query results based on relationships.

OSNs can have hundreds of millions of users, and each user can have tens of devices. The underlying database querying interface has to scale to this large number of devices. Distributed databases such as Cassandra [9] support transparent scaling. Graph databases such as Neo4J [21] are optimized to perform queries involving native graph operations faster. Depending on the type of query, schema-free relational databases and graph databases may outperform each other. We envisage that designing a highly scalable, available and fast database for systems like SBone will be an important database challenge.

4.5 Device Pairing

Once devices find each other, the devices perform secure pairing. Unlike the traditional definition of pair-

ing, device-to-device pairing in SBone refers not only to near-field communication, but also to the case when the devices communicate via internet. Devices exchange their public key certificates with each other. SBone uses the public keys stored in the SBone server to perform device-to-device pairing.

4.6 Device State Sharing

SBone enables users to share the state information of their personal devices with their friends. If the state of a device is popular, repeated queries may place undue strain on the device's resources. Devices have limited battery, and communication bandwidth. When repeated queries are placed on a device, the results of the query are cached on the SBone server. SBone maintains a staleness indication to indicate how fresh the state information is, using, for example, the last updated timestamp.

4.7 Resource Sharing

SBone enables users to share the resources of their personal devices, such as internet connectivity, GPS, camera, and talktime, with devices belonging to their friends. Some resources can be used simultaneously by multiple users, such as internet connection. Other resources can be used by only one user at a time, such as talk time.

For users to share their resources with others, several user concerns need to be addressed. Firstly, when the owner wants to use the sharable resource, SBone must implement a Quality of Service (QoS) mechanism that allows the user to control the level of sharing.. Secondly, in the case of a resource that cannot be simultaneously accessed by multiple users, SBone must implement a scheduling scheme using which users can reserve resources. Finally, not all users may be altruistic in sharing. Some users may require an incentive model to encourage them to share their resources. To implement any incentive mechanism, SBone must have an accounting component that logs the resource usage.

Owner Priority: The owner must be able to control the level of sharing of the resource. When the owner accesses the shared resource, SBone should give priority to the owner. Further, the owner must have the ability to temporarily disable sharing for the resource, whenever she wants sole access to it.

Reservation: If resource cannot be simultaneously accessed by multiple devices, friends need to have a mechanism using which they can reserve access to a resource. A scheduler running on the device ensures that the reservations are followed. We assume that the owner also has to follow this reservation.

Accounting: The accounting module logs the resource usage so that, optionally, a billing scheme can be incorporated. Different metrics can be used, such as time of usage, or a resource specific metric (like bytes downloaded for an internet resource, or pictures taken for a camera resource).

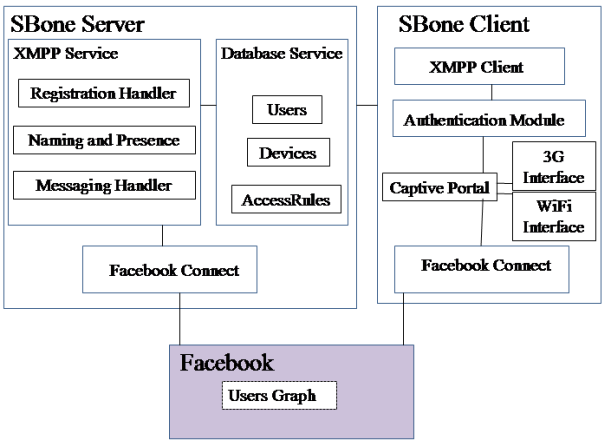


Figure 4: **SBone Prototype components. The prototype consists of three components, the SBone server, the SBone client and the SBone Facebook application.**

5 Case Study

We implemented an SBone prototype that allows personal mobile devices to connect to the internet using the internet resource of devices that belong to their friends. The scenario corresponds to Figure 3(a), where the client devices A and B are Linux laptops, the SBone server is a Linux server, and the social network is Facebook. A has a sharable internet resource. Our prototype enables B to get connected to the internet using A's connectivity.

5.1 SBone components

The SBone prototype consists of three components, the SBone server, the SBone client and the SBone Facebook application, as illustrated in Figure 4. The server manages registration, naming and discovery. The SBone client performs the device sharing lifecycle consisting of the registration, connection and sharing phases. The Facebook application is used for authentication and access control.

SBone Server. The SBone server maintains the network of users and devices using an Extensible Messaging and Presence Protocol (XMPP) [33] server. XMPP is an open, XML-based protocol for message-oriented middleware such as Instant Messaging (IM), Voice Over IP (VOIP) and file transfer signaling. We chose XMPP since the protocol already provides mechanisms for nam-

ing, presence and messaging, in addition to other services like video and audio. Popular deployments like Google's GTalk are testaments to XMPP's scalability. Due to the use of XML, the protocol is extensible. We used Jabberd2, an open-source XMPP server. The SBone server stores the user and device information in a Mysql database, which is accessed by the XMPP server.

SBone Client. The SBone clients run an XMPP client, which registers to the SBone server using the owner's Facebook credentials. After registration, the connects and client sends the device's routable name to the server, to allow other clients to reach it. The client also sends its list of resources shared to the server.

SBone Facebook Application. The SBone Facebook application allows users to visualize their device network. Figure 5 shows a screenshot of the SBone Facebook application. Users can see the list of devices owned by them, and their friends. A device that is currently connected to the internet is displayed as *online*. Users can send messages to devices using this interface. Devices can also directly send messages to each other. The XMPP protocol supports XML messages, so our prototype can be easily extended to share pictures, audio and video.

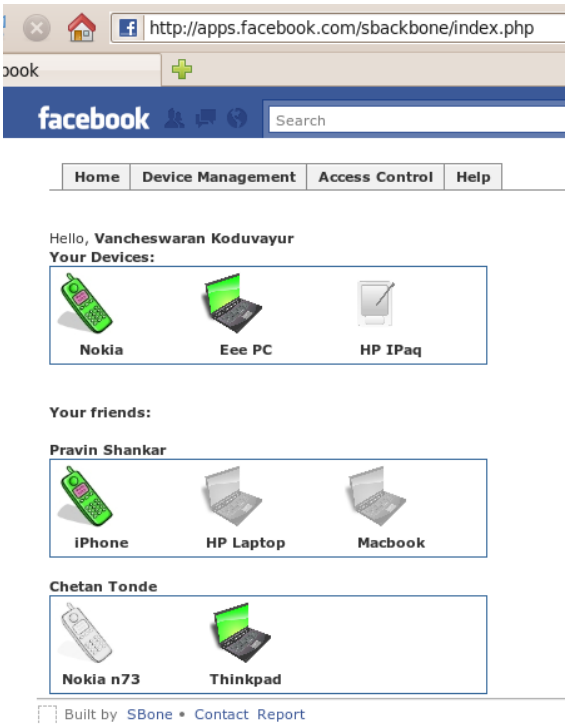


Figure 5: **SBone Facebook application. Users can see the online status of their devices, and the devices owned by their friends.**

5.2 Internet Connectivity Sharing

We implemented internet connectivity resource sharing functionality between devices A and B, as described in Figure 3(a), using SBone. A is a Linux laptop that runs an SBone client, with the WiFi interface set up as an access point, to provide internet connectivity sharing to devices in its social network. Internet connectivity sharing is implemented using a simple captive portal, which we built using *iptables*. A captive portal intercepts normal network traffic from a newly associated wireless client, B, and diverts it to a website where B can authenticate. Using *iptables* filtering rules, all IP traffic is diverted to a web application running on A.

For authentication, the captive portal uses Facebook Connect [11]. Facebook Connect is a set of APIs provided by Facebook, which allow a Facebook user to login to the SBone server, an external website, using Facebook credentials. Once authenticated, the user's identity and social connections are available to SBone (with user consent). Once B logs in, the server retrieves U_B's Facebook friends list. If U_A and U_B are friends, SBone grants access to B by adding an exception in A's filtering rules.

6 Discussion and Future Work

Sociology of sharing. Sociologists have studied human relationships, and the factors that influence sharing between humans [15]. With the advent of social media, people have analysed how users share news and personal data with friends on social networks [20, 17]. Given a set of personal devices, it is important to study the structure of networks formed by sharing patterns of these devices. Once people start sharing state and resources of their personal devices with friends, we envisage that this will become an important research area that combines social networks and computer networks.

Privacy. Devices possess personal information about users, that they may not wish to share with others. A naive solution for state sharing can leak personal information of people to unintended users. Even if the system provides access control policies, users may find it difficult to specify fine-grained access control for their device state, and are prone to making mistakes. Designing better and more usable privacy controls for social networks is an important problem, which becomes even more relevant as users share state of their personal devices with each other.

Inferring trust from social relationships in an OSN is a hard problem. Recently, Google Buzz [8] introduced an auto-follow feature, which inferred that users may want to share their personal state with their frequent email contacts. This feature raised privacy concerns, since frequent email contacts could be co-workers or other people

with whom one may not necessarily want to share their personal state. Privacy concerns have to be addressed for people to share personal state of their devices. As future work, we will investigate better ways to infer levels of trust between users based on their social relationships in the social networks.

Incentive Models. Not all users may be altruistic in sharing their resources with friends. Some users may require an incentive model to encourage them to share their resources. Moreover, in the case of resources that depend on service provider's cooperation, such as internet access, or sharing talktime, the incentive model should provide an incentive to the service provider as well. We plan to investigate incentive models for resource sharing in personal devices.

Secure Internet Sharing. Our internet resource sharing application makes use of an http captive portal to authenticate users. This solution works for typical home users, but may not satisfy the security requirements of an enterprise setting, where wireless networks require to be encrypted. A commonly used enterprise wireless setup consists of WPA2 Enterprise mode, with 802.1x authentication using a RADIUS server. As future work, we plan to enhance our prototype to support enterprise settings, using a modified RADIUS server that uses a social network to authenticate clients.

Different Types of Social Networks. In this paper, we do not limit our system to any specific social network. SBone can leverage social relationships from any existing online social network. However, not all OSNs are the same. People use Facebook to connect with their friends and acquaintances, whereas they use LinkedIn to connect with their professional contacts. Celebrities use sites like Twitter or Facebook Pages to connect with their followers. A link between two people can indicate different levels of trust on different OSNs. As future work, we will explore ways to infer the semantic meaning of social links based on the OSN.

7 Conclusion

We proposed an architecture that allows personal devices to share their resources and state with each other, using a social network for authentication, naming, discovery and access control. We discussed the design challenges introduced by SBone, and presented a case study describing how SBone can be used to enable internet sharing between mobile personal devices.

References

[1] ALLMAN, M. Personal namespaces. Proceedings of ACM SIGCOMM HotNets Workshop.

[2] ALLMAN, M., RABINOVICH, M., AND WEAVER, N. Relationship-oriented networking. ICSI Networking Group Technical Report.

[3] ASOKAN, N., MOLONEY, S., GINZBOORG, P., AND KOSTIAINEN, K. Visitor access management in personal wireless networks. Proceedings of the Seventh IEEE International Symposium on Multimedia.

[4] BALFANZ, D., DURFEE, G., GRINTER, R. E., SMETTERS, D., AND STEWART, P. Network-in-a-box: How to set up a secure wireless network in under a minute. Proceedings of the Usenix Security Symposium.

[5] BALFANZ, D., SMETTERS, D. K., STEWART, P., AND WONG, H. C. Talking to strangers: authentication in ad-hoc wireless networks. Proceedings of Network and Distributed System Security Symposium (NDSS).

[6] BANKS, L., YE, S., HUANG, Y., AND WU, S. F. Davis social links: integrating social networks with internet routing. Proceedings of the ACM SIGCOMM workshop on Large scale attack defense, pp. 121-128.

[7] BURKE, J., ESTRIN, D., HANSEN, M., PARKER, A., RAMANATHAN, N., REDDY, S., AND SRIVASTAVA, M. B. Participatory sensing. Proceedings of World Sensor Web Workshop, ACM Sensys.

[8] Google Buzz. http://www.google.com/buzz.

[9] The apache cassandra project. http://incubator.apache.org/cassandra/.

[10] DOURISH, P., GRINTER, R., DE LA FLOR, J. D., AND JOSEPH, M. Security in the wild: User strategies for managing security as an everyday, practical problem. Personal and Ubiquitous Computing, 8(6), 391-401.

[11] Facebook Connect. http://developers.facebook.com/connect.php.

[12] Facebook statistics. http://www.facebook.com/press/info.php?statistics.

[13] FORD, B., STRAUSS, J., LESNIEWSKI-LAAS, C., RHEA, S., KAASHOEK, F., AND MORRIS, R. Persistent personal names for globally connected mobile devices. Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI).

[14] FORD, B., STRAUSS, J., LESNIEWSKI-LAAS, C., RHEA, S., KAASHOEK, F., AND MORRIS, R. User-relative names for globally connected personal devices. Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS06).

[15] GRANOVETTER, M. The strength of weak ties. Proceedings of the American Journal of Sociology, 78 (May): 1360-1380.

[16] HIROFUCHI, T., KAWAI, E., FUJIKAWA, K., AND SUNAHARA, H. Usb/ipa peripheral bus extension for device sharing over ip network. Proceedings of USENIX Annual Technical Conference, FREENIX Track, pp. 47-60.

[17] KEMPE, D., KLEINBERG, J., AND TARDOS, E. Maximizing the spread of influence through a social network. Proceedings of the 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining.

[18] KONG, J., GANEV, I., SCHWAN, K., AND WIDENER, P. Cameracast: Flexible access to remote video sensors. Proceedings of Multimedia Computing and Networking (MMCN).

[19] KWON, W., CHO, H. W., AND SONG, Y. H. Design and implementation of peripheral sharing mechanism on pervasive computing with heterogeneous environment. Software Technologies for Embedded and Ubiquitous Systems, Volume 4761.

[20] LESKOVEC, J., BACKSTROM, L., AND KLEINBERG, J. Memetracking and the dynamics of the news cycle. Proceedings of the 15th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining.

[21] neo4j. http://neo4j.org/.

[22] OpenID Foundation. http://openid.net/.

[23] Information and Communication Technology (ICT) statistics. http://www.itu.int/ITU-D/ict/.

[24] PUTTASWAMY, K. P., AND ZHAO, B. Y. Preserving privacy in location-based mobile social applications. Proceedings of HotMobile Workshop.

[25] RAMACHANDRAN, A., AND FEAMSTER, N. Authenticated out-of-band communication over social links. Proceedings of the First ACM SIGCOMM Workshop on Online Social Networks.

[26] ROSENBERG, J., MAHY, R., AND MATTHEWS, P. Traversal Using Relays around NAT (TURN). Internet-Draft.

[27] ROSENBERG, J., MAHY, R., MATTHEWS, P., AND WING, D. Session Traversal Utilities for NAT (STUN). RFC 5389.

[28] SASSE, M., BROSTOFF, S., AND WEIRICH, D. Transforming the weakest link a human/computer interaction approach to usable and effective security. Proceedings of BT Technology Journal, 19 (3). pp. 122-131.

[29] SHEHAN, E., AND EDWARDS, W. K. Home networking and HCI: what hath god wrought? Proceedings of the SIGCHI conference on Human factors in computing systems.

[30] TOOTOONCHIAN, A., SAROIU, S., GANJALI, Y., AND WOLMAN, A. Lockr: Better privacy for social networks. Proceedings of the Fifth ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT).

[31] WiFi Protected Setup (WPS). http://www.wi-fi.org/wifi-protected-setup/.

[32] WU, X., WANG, W., LIN, B., AND MIAO, K. Composable io: A novel resource sharing platform in personal clouds. Proceedings of 1st International Conference on Cloud Computing.

[33] XMPP protocol specification. http://xmpp.org/tech/overview.shtml.