MoRePriv: Mobile OS-Wide Application Personalization

Drew Davidson
University of Wisconsin

Benjamin Livshits
Microsoft Research

Abstract

We present MoRePriv, a system that combines the goals of privacy and content personalization in the browser. MoRePriv discovers user interests and shares them with third-parties, but only with the explicit permission of the user. We demonstrate how always-on user interest mining can effectively infer user interests in a mobile operating system. Inference is based on parsing and classifying multiple stream of (sensitive) information about the user, such as their email, SMS, Facebook stream, and network communications.

This paper positions MoRePriv as an OS-level service that exposes information about the user in the form of a *persona* to applications. This provides easy-to-use APIs and limits the potential for information leaks. Using a number of cases studies, we demonstrate how various personalization tasks can be achieved with the help of MoRePriv.

1. Introduction

Whether we like it or not, today traditional PC applications geared towards the end-user such as Microsoft Word, Adobe Reader, email readers, and many others may seem downright boring and inflexible. These applications are *static* — while some customization opportunities exist, they work more or less the same no matter who is using the computer, what the prior history of application use is, and so forth. In contrast, the web applications and mobile apps that are rapidly gaining popularity among users are *dynamic* — they change to suit the user's needs and preferences in response to observed interactions. We think that it is inevitable that traditional applications will learn from this trend, and eventually follow suit.

As more applications adopt *personalization*, the question arises as to where the data that drives this functionality comes from. Currently, personalized applications in the web and mobile domains collect this data on an individual, ad-hoc basis. However, as we argue in this paper, it is to the benefit of users and application developers to draw this information from a single, unified, consistent source. This trend provides an opportunity for operating system developers, especially in the context of mobile devices. On smart handhelds, we argue, operating systems evolve rapidly, allowing for game-changing innovation. In this paper, we

- examine some noteworthy examples of personalization in the web and mobile space;
- describe the "players" in this new space and explain the forces driving personalization;
- illustrate key opportunities for mobile operating system designers and propose specific ways in which applications can be enhanced to become more personalized;
- discuss challenges that must be overcome in order for a personalization mechanism to gain widespread adoption, such as privacy concerns and performance impact.

1.1 Paper Organization

The rest of this paper is organized as follows: Section 2 gives some illustrative examples of personalization on the web and in mobile applications. Section 3 points out new opportunities for personalization in the context of mobile operating systems. Section 4 discusses privacy and performance implications. Section 5 concludes the paper.

2. Overview

Our previous work has explored personalization opportunities in the context of a web browser. The basic idea is to collect an ever-updating *user interest profile*, based on the user's browsing history, by classifying the sites the user visits. This approach has been implemented in a system called RePriv [4], allowing for highly accurate

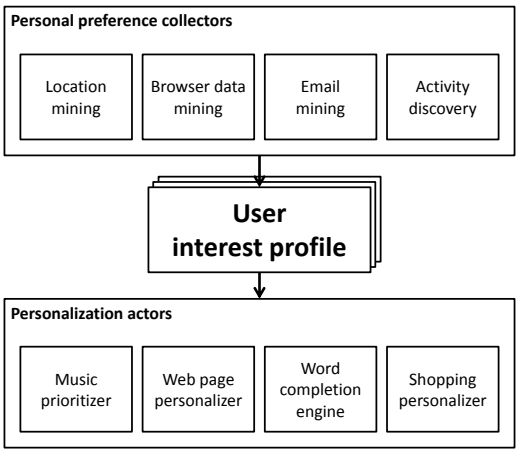


Figure 1. Analysis architecture.

The Federal Trade Commission last week released a report calling for a "do not track" system that would allow people to send a message through their Web browsers alerting tracking companies that they don't wish to be tracked.

Microsoft said its coming tool is potentially more powerful than a do-not-track system that relies on companies to comply with a user's request."This path is different in that it actually blocks the tracking now," said Dean Hachamovitch, Microsoft's vice president in charge of Internet Explorer development. He added the two types of blocking could "happily both coexist."

FTC Chairman Jon Leibowitz applauded Microsoft's move and called on other makers of Web browsing software to offer similar features.

(a) Original text.

The Federal Trade Commission last week released a report calling for a "do not track" system that would allow people to send a message through their Web browsers alerting tracking companies that they don't wish to be tracked.

Microsoft said its coming tool is potentially more powerful than a do-not-track system that relies on companies to comply with a user's request." This path is different in that it actually blocks the tracking now," said Dean Hachamovitch, Microsoft's vice president in charge of Internet Explorer development. He added the two types of blocking could "happily both coexist."

FTC Chairman Jon Leibowitz applauded Microsoft's move and called on other makers of Web browsing software to offer similar features.

(b) Important text highlighted.

The Federal Trade Commission last week released a report calling for a "do not track" system that would allow people to send a message through their Web browsers alerting tracking companies that they don't wish to be tracked.

Microsoft said its coming tool is potentially more powerful than a do-not-track system that relies on companies to comply with a user's request." This path is different in that it actually blocks the tracking now," said Dean Hachamovitch, Microsoft's vice president in charge of Internet Explorer development. He added the two types of blocking could "happily both coexist."

FTC Chairman Jon Leibowitz applauded Microsoft's move and called on other makers of Web browsing software to offer similar features.

(c) Entity extraction and color.

Figure 2. Mobile text view.

personalization opportunities, including site personalization, precise ad targeting, and personalized search.

We believe that in the context of the mobile operating system, all of these opportunities exist as well as many others.

This is simply because at the level of the operating system, many built-in services and mobile apps other than the browser become open to instrumentation. With this increased information content, resulting in more *signal* to process, part of the challenge is in determining what information is relevant. For instance, is email content being sent to the user always relevant? Clearly, this is not the case for spam, or even rarely-read mailing list subscriptions. Are mp3 files stored on the mobile phone always representative of the user's music tastes? Is their mobile "check in" history representative of their restaurant tastes?

2.1 System Architecture

As shown in Figure 1, we envision a system in which both applications and the underlying operating system expose opportunities for personalization. User interactions are observed by *collectors* (top of the figure) and compiled to a user interest profile, which is subsequently used by *actors* for different forms of personalization (bottom of the figure).

As an example, the mobile OS can expose notification APIs for email arrivals, Facebook friend check-ins nearby, or activity discovery such as whether the user is walking or driving. The former can be processed and the file contents classified using a text classifier into Open Directory Project (ODP) categories. The latter would require a location-based classifier to provide the user with location-based recommendations.

It remains to be seen whether non-OS level app interactions provide a strong enough signal for acquiring user interests. For instance, is analyzing a given app's communication with a backend REST service and running this data through a text classifier going to provide useful data about user's preferences?

2.2 Advantages

While some of these ideas have been explored individually, we believe that the architecture we are proposing shown in Figure 1 has the following benefits:

- A single OS-level data source for personalization (as opposed to application-specific, ad-hoc information sources) allows seamless and uniform functionality for a single user across many applications, devices, and platforms.
- Across-the-board data collection for a single repository allows richer and more extensive data to be collected.
- Drawing data for personalization from a single, OS-controlled source makes the privacy issue more tractable. This is because a single, unified policy language can apply to all applications, and privileged OS-level mechanisms can aid in the enforcement of these policies (e.g. with the use of DIFC). Additionally, all data used for personalization is housed in a single place, reducing cognitive burden on the user when it comes to reasoning about privacy policies.

These ideas are especially powerful in mobile operating systems such as iOS, Android or Windows Phone, since people tend to entrust a fair bit of rich personal information to various mobile apps and because of various context awareness signals such as location, accelerometer, NFC, etc. This is also the very reason that privacy controls are clearly needed.

2.3 Classification

An approach to unifying personalization efforts is to provide a common taxonomy of user interests.

ODP: One such commonly used taxonomy is the ODP - the Open Directory Project [15]. A sample interest profile could be the following collection of interests: The

Science: Astronomy: Galaxies: Milky Way	.2
Sports: Hockey: Ice Hockey: Women	.7
Computers: Security: Internet: Privacy	.6

number indicates the degree of certainly in a particular interest. This taxonomy can be used to represent user interests based on their browsing history or the documents they type using a note taking app. Adnostic [13] is an example of a project using ODP or other similar taxonomies.

Custom interest profiles: Note that the one-size-fits-all approach does not quite work for every type of information. For instance, click information can be used to personalize menus in complex applications. This, obviously, falls outside of the ODP taxonomy and is better served with a custom interest profile. An example of this is a restaurant recommendation app such as Trumpet or a personalized news magazine such as Flipboard, or Zite.

2.4 Applying Personalization

We envision a number of templates for personalization. Some include reordering or pruning long menus in a mobile app UI, proposing related locations such as restaurants, and augmenting text display to fit user's tastes on a small screen as shown below.

2.5 Storage

While the default storage strategy is to keep the interest profiles local, on the current device, it is entirely possible to synchronize them - in an encrypted form - with the cloud. This is not unlike the approach used in Firefox Sync[18] for synchronizing browser data across multiple browsers. In addition to synchronization across multiple devices, some desktop, some mobile, some tablets, cloud synchronization also serves as a backup. Precedents of this exist in several domains, including bookmark synchronization, Microsoft Office setting synchronization, Dropbox[19], automatic note synchronization with Windows Mobile phones and Windows Live, etc.

Figure 3. Personas.

3. Implementation

In order to test the effectiveness of personal preference collectors, we instrumented Windows Phone 7 to capture *personalization signals*, sources of data that indicate the preferences of the user. We then use these personalization signals to classify the user according to a number of *personas* on the device, which comprise a limited user interest profile. We then envision two facilities for personalization actors in MoRePriv: a privileged service to perform automatic personalization within the OS, and a set of APIs that give third party applications to the user interest profile.

3.1 Personas

Personas are custom interest profiles that represent various walks of life, offering different targets for personalization. In our prototype implementation of MoRePriv, we target 8 personas, listed in Figure 3. Each persona is represented by a Baysian classifier, trained on a manually curated list of keywords characteristic to that profile. For example, the Executive per-

sona represents strong interest in business, finance, and national news. Thus, the corresponding classifier is populated with text from such sites as the Financial Times. On the other hand, the Technophile profile represents a strong interest in technology, so the corresponding classifier is populated with text from tech blogs.

While we believe these profiles are a reasonable proof of concept, we note that our system is modular with respect to the profiles that are used, and the system could easily by modified by training a Baysian classifier on a new list of keywords.

Personas are archetypes of interests. In practice, no user is likely to have interests that exactly one persona. As such, each persona is assigned a relevance score that indicates how closely the interests of the user are matched by that persona. Consider an example user with a high relevance score to the Technophile persona and a moderate relevance score to the Executive persona. This person might be very interested in technology news, and somewhat interested in financial news.

3.2 Personalization Signals

In order to assign relevance scores to each persona, MoRePriv needs data to classify. Here, we leverage our position at the OS level: all user information must pass through the operating system in order to be consumed or produced by the user. In our implementation, we capture five distinct personalization signals, as listed in Figure 4.

The Email, Facebook, and Twitter signals are captured by modifications to the C# core classes within the Windows Phone framework, upon which apps are built. The SMS and Network interface signals are captured by modifying the Windows Phone OS itself. These signals demonstrate an advantage of performing signal capture at the Operating System level: since the OS and framework have a very high level of privilege, the user must already trust these components to handle personal data. As such, the signal capture mechanisms are already within the user's trusted computing base.

Furthermore, instrumentation at the OS level has the unique advantage of being able to integrate multiple data sources together. This is important for several reasons. Firstly, even very rich data sources can suffer from a cold-start problem, but are useful in aggregate. Figure 5 shows the Technophile relevance scores for the example user with a strong interest in technology. The "wall" line represents the relevance score as posts on the user's Facebook wall are added. The "posts" line represents the relevance score as posts made my the user are added. The "likes" line represents the relevance score as the user "likes" additional things. In all three of these cases, it takes a fair amount of data for the signals to converge.

Source	Description
Email	Bodies of all incoming and outgoing email messages
Facebook	Facebook "likes", posts by the user, posts on the user's wall
SMS	Incoming and outgoing text messages
Twitter	Tweets posted to the user's feed
Network interface	All other http traffic to and from the device

Figure 4. Personalization data sources explained.

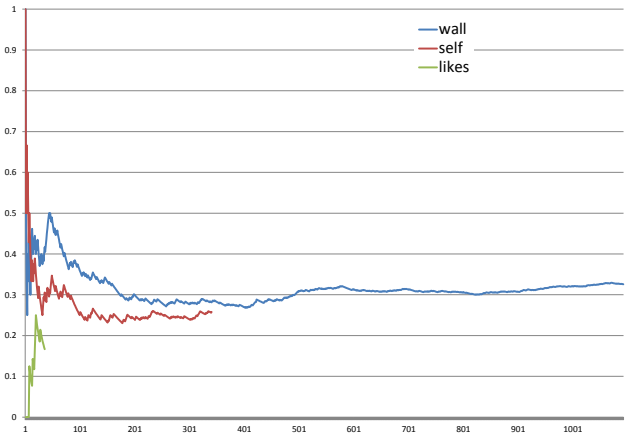


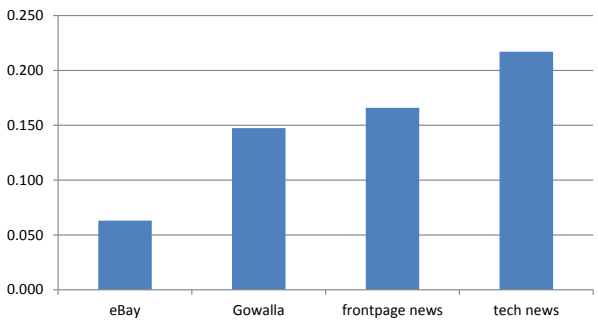
Figure 5. Facebook data convergence.

Figure 6 lists user relevance scores that would be assigned by the Technophile classifier input across different sites to the same user. As shown in Figure 6a, the quality of signals in classifying a user varies significantly. However, as shown in Figure 6b, combining these signals together can boost the correct relevance score even in the face of highly irrelevant signal data, such as data from EBay.

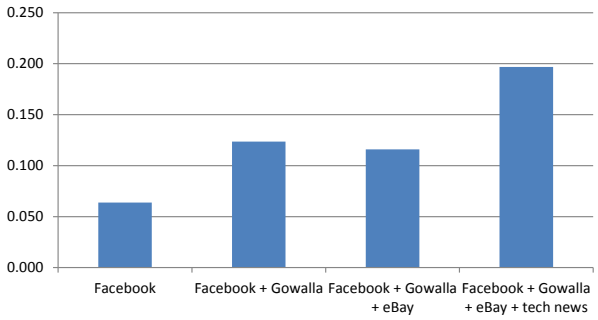
In MoRePriv we give the user the option to switch data collection on and off, but we envision a use case in which the data collection is always on, refining each persona's relevance score as the user interacts with their mobile device.

3.3 OS-level Service

Positioning MoRePriv within the OS provides an opportunity to collect a great deal of data to build a user interest profile. However, it also provides an opportunity to perform personalization on user-level apps without any modification of the app itself. To explore the possibility of automatic personalization, We altered the Windows Phone C# framework to alter the way GUI elements are displayed to the user based on the persona



(a) Sources compared.



(b) Sources combined.

Figure 6. Sources of personalization data.

relevance scores. For example, we altered the order in which elements of unsorted lists are displayed on the screen. For legacy applications such as news readers, this had the effect of not only reordering the order in which stories are displayed (stories more relevant to the user's interests are shuffled to the top), but it also has the effect of reordering entire categories of subjects such that, to continue our running example, the tech page of a news reader app appears earlier in the menu than the arts page when OS-level personalization is active.

3.4 MoRePriv APIs

MoRePriv also exposes APIs to third party developers that allow application-specific personalization. There are 5 API functions listed below. These APIs can be accessed via a user-mode library that can be bundled with an app. Consequently, a single app can be written to the MoRePriv APIs that will work on a standard image of the Windows Phone without the MoRePriv enhancements (albeit without personalization):

- `IsMoRePrivImage()` returns true if the operating system is built with MoRePriv personalization enhancements
- IsMoRePrivEnabled() returns true if personalization is enabled. We allow users to toggle personalization on and off.
- `Classify(String s, Object o)` classifies the relevance of `o` to persona `s`. For example, `Classify("Technophile", "Computer")` will return a high value, because computers are of high interest to Technophiles. Note that this call does not reveal any information about the user to the app, it is simply a convenience method to allow apps to classify objects.
- `TopProfile()` return the most relevant profile to the user if personalization is enabled, and `null` if personalization is disabled.
- `Ignore(Object o)` Informs MoRePriv not to apply OS-level personalization to `o`. This allows developers to bypass the GUI features such as automatic list reordering.

4. Case Studies in Personalization

This section covers several common uses of personalization we see.

4.1 Personalizing Browsing and Display

In the RePriv project, we discuss ways in which the browsing experience can be personalized. Switching focus to the mobile context, where browsers are not nearly as mature, we outline how a typical mobile browser can be augmented with personalization.

- Text summarization. An example of this is taking a long Wall Street Journal article and trying to read it on a typical mobile device screen. As shown in Figure 2, by default, the article is truncated and involves a lot of scrolling to get to the contents. An alternative is a smart personalized summarization technology that highlights the parts of the text likely to be important for a given user and puts the rest under ellipses. Another component of this is highlighting extracted entities and perhaps embedding links to them, as shown in Figure 3.
- Form fill enabled. This speaks to the idea of a *digital wallet* that is maintained on one's phone.
- Word completion while typing. For example, having typed `decla`, the completion might be `declaration` for a user interested in *law* and `declamation` for a user interested in *public speaking* or *opera*.
- Spell checking with a custom user-specific dictionary.
- "Recent documents" file open menu ordered by frequency of use.
- Suggested web sites based on the user's browsing history.

- Extensions, such as GetGlue, can draw personal information directly from the operating system, thus simplifying the extension implementation while providing richer data.

4.2 RSS News Feed Personalization

To test the usefulness of the MoRePriv APIs, we built a custom RSS reader called MoRSS. This app pulls stories from 10 RSS news feeds, and samples from this news feeds to display a list of stories to the user. MoRSS disables the OS-level GUI enhancements described in Section 3. Instead, MoRSS relies on the built-in table in Figure 7 to rate how interesting each of the RSS feeds that MoRSS subscribes to will be to a profile.

MoRSS can operate with no personalization, in which case stories from each RSS feed will be sampled uniformly and displayed to the user in the order in which they are sampled. When personalization is enabled, MoRSS queries the MoRePriv API to determine the top profile of the user, and then samples according to the column of Figure 7 for that persona.

Figure 8a shows an example screenshot of MoRSS with no personalization applied. In Figure 8b, the same set of stories are sampled according to the interests of the Soccer Mom persona column of Figure 7, which places an emphasis on Health and Entertainment stories. Similarly, Figure 8c shows the same set of stories sampled by the interests of the Technophile column.

This demonstrates two advantages of exposing limited information to third party applications:

1. developers have the flexibility to reinterpret the top profile in any way that they see fit. Apps such as MoRSS are free to samples tech stories for the Soccer Mom, even though the built-in Baysian classifier for that profile does not have tech keywords.
2. it allows nuanced, fine-grained personalization. Developers can use personalization in whatever way they see fit, operating over objects such as RSS feeds or images according to their own understanding of how well that object fits with a persona. Furthermore, the personalization can be done in a privacy-preserving way: MoRSS uses client-side personalization, so even the owner of the RSS feeds cannot learn the top profile of the user from the requests that MoRSS makes.

4.3 Non-Networked Personalization

Though we have focused primarily on personalization as it related to networked device use, OS-level personalization has broad applicability. To demonstrate these possibilities, we implemented a simple calculator using the MoRePriv API. Figure 9a displays the default calculator with no personalization.

When personalization is enabled and the top profile is a Tween, the calculator is whimsically re-skinned for a pre-teen girl, as shown in Figure 9b.

When the top profile is a retiree, the same calculator goes into a high contrast, high usability mode in which text size is increased. MoRePriv provides an alternative to providing complicated configuration menus to users who nonetheless prefer different configurations. Although the calculator will perform personalization at each run, an alternative would be to use user profile data to provide an initial configuration that is likely to be close to what the user wants, and allow her to tweak configuration options from there.

5. Related Work

5.1 Privacy and Web Applications

As a reaction to the decrease in privacy on the web, many have started exploring techniques that can be applied to restore some degree of privacy while still allowing for the rich web applications that people have come to expect. The P3P Project [2], sponsored by the W3 Consortium, is an attempt to formalize and mechanize the specification and distribution of privacy policies on the web. It does not have provisions for providing personal information to content providers, however, making the issue of providing personalized functionality rather difficult. Jakobsson *et al.* [9] considered the problem of third-party sites mining users' navigation history. They developed a system that allows third parties to learn *aggregate* information about users' navigation histories, rather than the full listing. All privacy assurances offered by this system derive from the fact that its mechanism is easily auditable by end-users, so parties who wish to mine history data have disincentive to cheat.

Becker and Chen [1] found that it is possible to deduce specific personal characteristics given only a list of their friends on a social network. Worse yet, they found that it is very difficult to defend against this type of inference, assuming an attacker has access to the user's entire social graph: on average, they found that users would have to remove hundreds of friends from their connections in order to ensure the privacy of their own characteristics.

Narayanan and Shmatikov [14] studied the privacy implications of social network participation. Their observation is that the operators of online social networking sites now share user data with third parties, but only scrub personally-identifying information in an ad-hoc fashion. They developed a *re-identification* algorithm that relates users' privacy in a social network to node anonymity in the social network graph, and attempts to identify particular users from scrubbed social network data. They found that if a user subscribed to both Twit-

	activist	bachelor	business executive	football dad	retiree	soccer mom	technophile	tween
Health	3	3	4	4	10	7	2	1
Tech	4	6	5	5	3	4	10	8
US	9	4	7	6	6	4	3	2
Business	4	5	10	5	5	2	6	1
World	7	2	4	1	2	2	2	1
Entertainment	0	7	3	4	4	6	5	5
Science	2	3	3	1	3	2	6	4
Society	6	3	2	2	3	3	2	3
Politics	10	4	6	5	5	3	2	1
Sports	0	7	4	10	4	2	5	7

Figure 7. News personalization: parameters.

(a) No personalization. (b) Personalization (Soccer Mom). (c) Personalization (Technophile).

Figure 8. RSS news reader.

ter and Flickr, then the algorithm can correctly identify them with 88% accuracy.

McSherry and Mironov [12] attempted to restore a certain degree of privacy to collaborative recommendation algorithms, such as those used by Netflix and amazon.com. Citing the work of Narayanan and Shmatikov [13] in de-anonymizing users who take part in such systems, they worked in the framework of *differential privacy* [3] to build a an algorithm that preserves the privacy of each individual rating entered by a participating user. The performance is comparable to that of the original Netflix recommendation algorithm.

5.2 Privacy in Advertising

One problem that has received much recent attention is that of delivering targeted advertisements to web users without violating their privacy. Freudiger *et al.* [5] observe that the prevalent mechanism for targeting advertisements to individual users is the *third-party cookie*.

They propose a browser extension that allows users to directly manage third-party cookies in order to decide the degree to which advertisers are able to track them. However, unlike with MoRePriv, this solution does not give users arbitrary, fine-grained control over the type of information that is given to third-parties. Furthermore, advertisers have no incentive to obey the privacy safeguards instantiated by this mechanism. In a

(a) No Personalization (b) Personalization (Tween) (c) Personalization (Retiree)

Figure 9. Calculator personalization.

slightly different vein, several recent systems [6, 10, 17] attempt to remedy the problem by storing the necessary sensitive personal data on the client, along with all possible ads in the network. When an ad is displayed, it is matched to personal information locally, thus sidestepping the need to leak to the ad network. Accounting and click-fraud prevention are addressed using either additional semi-trusted parties, or homomorphic encryption. The primary difference between these systems and MoRePriv is generality: MoRePriv asks the user to provide content providers (in this case, an advertising network) with small amounts of selected personal data in return for full application generality, whereas these tools effectively hide all personal data needed to drive the single application of targeted advertising.

[?]

5.3 Managing Private Browser State

A number of researchers have studied ways to identify users and preferences from browser interactions. Wondracek *et al.* [18] found that a subtlety in the W3C specification that allows browser history to be inferred can be leveraged to de-anonymize users of popular social networking sites. Jackson *et al.* [8] attributed the problem of history sniffing to the fact that browsers do not extend the same-origin policy to the history state leveraged in the attack. Recently, Mozilla has taken steps to prevent history sniffing [16], at the cost of breaking certain parts of the W3C specification. In a broader development, Eckersley [4] introduced a technique dubbed *browser fingerprinting*, wherein a large number of publicly-visible browser attributes are combined to produce an identifying string shared by only one in about 286,777 browsers.

Several researchers have approached the technical problem of maintaining user anonymity while browsing. Howe and Nissenbaum [7] created TrackMeNot, a Firefox extension that attempts to anonymize search behavior by periodically submitting random search queries to major search engines. McKinley [11] examined the privacy modes of popular browsers, as well as their ability to clear private state when directed by the user. She found that while some browsers do in fact clear private state when instructed, none of the browsers' privacy modes performs as advertised; each browser left some form of persistent state that could be later retrieved by web pages in different browsing sessions.

Web Personalization and Mining: The basis on which personalization is performed varies from application to application. Pierrakos *et al.* [15] surveyed the topic of mining users' behavior on a set of web services to infer information that will aid personalization. They found that almost all web personalization efforts fall into one of four broad categories: memoriz-

ing information for later replay, guiding the user towards likely relevant information, customizing content to match users' interests, and supporting users' efforts to complete tasks. MoRePriv is designed primarily to support the implementation of the second and third points, but it can be used to support aspects of all types of personalization.

6. Conclusions

This paper proposes operating system-level mechanisms that simplify building personalized applications. Our focus is primarily on mobile operating systems. We believe that at the level of the operating system these opportunities are largely untapped at the moment. This paper summarizes the opportunities and describes how both interest collection and personalization can be enabled at the OS level. It also presents some novel and relevant personalization examples in the context of mobile devices.

Acknowledgments

References

[1] J. Becker and H. Chen. Measuring privacy risk in online social networks. In *Proceedings of the Workshop on Web 2.0 Security and Privacy*, May 2009.

[2] W. Consortium. Platform for Privacy Preferences (P3P) Project. `http://www.w3.org/P3P`.

[3] C. Dwork. Differential privacy: a survey of results. In *Proceedings of the International Conference on Theory and Applications of Models of Computation*, May 2008.

[4] P. Eckersley. How Unique Is Your Web Browser? Technical report, Electronic Frontier Foundation, Mar. 2009.

[5] J. Freudiger, N. Vratonjic, and J.-P. Hubaux. Towards Privacy-Friendly Online Advertising. In *Proceedings of the Workshop on Web 2.0 Security and Privacy*, May 2009.

[6] S. Guha, A. Reznichenko, K. Tang, H. Haddadi, and P. Francis. Serving Ads from localhost for Performance, Privacy, and Profit. In *Proceedings of Hot Topics in Networking*, Nov. 2009.

[7] D. C. Howe and H. Nissenbaum. TrackMeNot: Resisting surveillance in web search. In I. Kerr, V. Steeves, and C. Lucock, editors, *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, chapter 23. 2009.

[8] C. Jackson, A. Bortz, D. Boneh, and J. C. Mitchell. Protecting browser state from web privacy attacks. In *Proceedings of the International Conference on World Wide Web*, May 2006.

[9] M. Jakobsson, A. Juels, and J. Ratkiewicz. Privacy-Preserving History Mining for Web Browsers. In *Proceedings of the Workshop on Web 2.0 Security and Privacy*, May 2010.

[10] A. Juels. Targeted advertising ... and privacy too. In *Proceedings of the Conference on Topics in Cryptology*, Apr. 2001.

[11] K. McKinley. Cleaning Up After Cookies Version 1.0. Technical report, ISEC Partners, Dec. 2010.

[12] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the net. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, Jun. 2009.

[13] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2008.

[14] A. Narayanan and V. Shmatikov. De-anonymizing social networks. *IEEE Sympolsium on Security and Privacy*, May 2009.

[15] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C. D. Spyropoulos. Web usage mining as a tool for personalization: A survey. *User Modeling and User-Adapted Interaction*, 13(4), 2003.

[16] The Mozilla Team. Plugging the CSS History Leak. `http://blog.mozilla.com/security/2010/03/31/plugging-the-css-history-le%ak`, 2010.

[17] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy preserving targeted advertising. In *Proceedings of the Network and Distributed System Security Symposium*, Feb. 2010.

[18] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, May 2010.