Privicons
A approach to communicating privacy preferences between Users*

E. Forrest, J. Schallaböck
Stanford/Berlin November, 2010

Introduction.

Alongside privacy challenges posed by technical problems like faulty architecture or insecure protocols are everyday privacy harms caused by basic failures of communication. For example, when a user unthinkingly forwards an e-mail chain that includes crass private remarks, or casually passes along information from an e-mail that was meant to have been kept secret, that user violates privacy by ignoring or misunderstanding norms, not code. Privicons uses a strategy of code-based norms,[1] or a "neighborliness" approach,[2] to address communications privacy problems like e-mail carelessness that occur within the bounds of code but nevertheless are ultimately problems of privacy norms and social signals: problems not readily solvable by code alone.

Purely code-backed strategies, like e-mail clients that refuse to allow certain messages to be forwarded or printed, tend to restrict speech indiscriminately. Such approaches can be overbroad and paternalistic toward users, or limited to certain platforms or applications. Privicons, however, uses an adaptable, open source strategy to place easily understandable privacy communications tools in users' hands, on any platform, without restricting what the user can do or say.

The code in Privicons thus serves mainly to help users clarify and understand social norms, signals, and expectations about privacy. If users can easily convey their privacy expectations, and recipients can understand and process those expectations in terms of widely understood social norms about privacy, then courtesy and care will help to prevent privacy harms caused by carelessness or misunderstanding about privacy expectations.

It is important to note that Privicons merely asks an e-mail's recipient to respect the sender's preference. Unlike, for example, DRM-oriented approches, our approach relies on social norms meant to be followed by the recipient, rather than technical enforcement mechanisms. However, developers who choose to create other applications to interact with Privicons – such as e-mail clients coded to recognize Privicons – might create new solutions that allow users to decide whether to opt-in to technical enforcement. Allowing Privicons' instructions to be overridden enables users to take control of their speech. Code here functions mainly to distribute Privicons' conceptual core.

Vocabulary

Figure: Draft Privicons including their respective ASCII-representation and a graphical symbol

---

The project's conceptual core is a vocabulary of icons – Privicons themselves – that communicate a user's privacy expectations, like "Don't Attribute," "Keep Internal," and "Keep Secret." These icons are meant to be implemented through lightweight, open source applications that affix the icons and their explanations to e-mails, per the user's preferences – like laundry instructions for your e-mail, or a Creative Commons for privacy. Privicons are easily visualized through either text or an image-based Privicons application. This visual flexibility gives Privicons wide adaptability across platforms and devices, ensuring easy applicability to all current and developing forms of visual electronic communication.

Specification

To allow for protocol- or standards-based privacy strategies to support Privicons in the future, we have disseminated a first-draft experimental RFC (Internet Draft). This Internet Draft provides a detailed outline on how developers should use Privicons in the e-mail environment[3]. Based on the Internet Draft's specifications, e-mail headers might eventually incorporate Privicons preferences, creating the possibility of different levels of code-backed responses for supporting clients that interpret these headers. Other options are possible: the Internet Draft means only to standardize the basic functionality of Privicons.

Specifically, the Internet Draft proposes a syntax and semantics for an extension of the Internet Message Format[4] (e-mail message), which would allow a Sending User of an e-mail to express his or her preferences on how the message content should be handled by the Receiving Users. For this purpose, the Internet Draft describes semantic sets of different character combinations – the Privicons in ASCII form – as outlined above, and describes how they may be used in a protocol- or standards-based setting.

These semantic sets can syntactically be integrated in the first line of the e-mail's body, in the e-mail's subject line, and/or in a dedicated header of any e-mail message. The specification also provides rules for handling conflicts amongst Privicons, effectively allowing for a smooth transaction and migration for clients and users intending to make use of Privicons in a code-oriented context. The applied specification would allow a user to type a Privicon in an e-mail's body or subject line – conforming clients would then interpret this information and, for example, include the relevant Privicons and attendant privacy information in the e-mail's header, or automatically insert additional information on the specific meaning of a Privicon in the e-mail's footer. A possibility for expressing this vocabulary in HTML-based e-mails using the Privicons' graphical symbols is also envisioned in the next draft: the Internet Draft is meant to be a uniform set of guidelines for any developer who wishes to incorporate Privicons into her project, thereby ensuring that Privicons remains coherent even as the project is widely and collaboratively distributed.

Implementation

The first such Privicons application, currently in development, is a Google Chrome extension that incorporates Privicons seamlessly into the Gmail user interface. We hope to expand this implementation to other e-mail and browser platforms, and perhaps to social networks like Facebook. As the first Privicons application reaches completion, the team hopes to proceed with tentative commitments from e-mail service providers to test Privicons more widely. We also plan to provide the code through open source channels in order to encourage adaptation and development on a horizontal model.

[1] The Power of Code-Backed Norms by Jonathan Zittrain, http://yupnet.org/zittrain/archives/20#70.
[2] Gelman, Lauren Amy, Privacy, Free Speech, and 'Blurry-Edged' Social Networks (November 1, 2009). Boston College Law Review, Vol. 50, No. 5, 2009.
[3] http://www.ietf.org/id/draft-koenig-privicons-00.txt
[4] RFC 5322