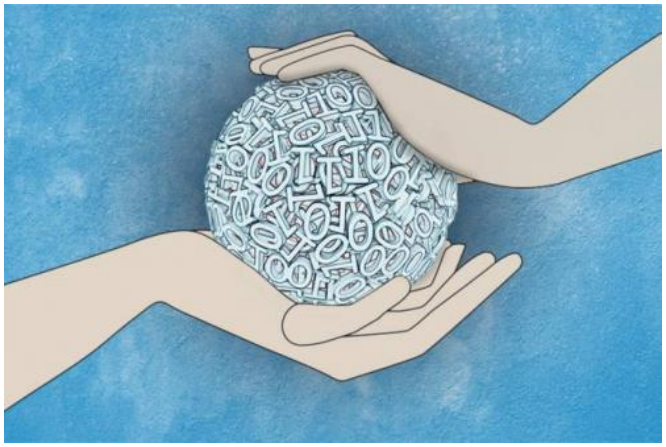
# New system would allow individuals to pick and choose what data to share with websites, mobile apps

9 July 2014



Credit: Christine Daniloff/MIT

Cellphone metadata has been in the news quite a bit lately, but the National Security Agency isn't the only organization that collects information about people's online behavior. Newly downloaded cellphone apps routinely ask to access your location information, your address book, or other apps, and of course, websites like Amazon or Netflix track your browsing history in the interest of making personalized recommendations.

At the same time, a host of recent studies have demonstrated that it's shockingly easy to identify unnamed individuals in supposedly "anonymized" data sets, even ones containing millions of records. So, if we want the benefits of data mining—like personalized recommendations or localized services—how can we protect our privacy?

In the latest issue of *PLOS ONE*, MIT researchers offer one possible answer. Their prototype system, openPDS—short for personal data store—stores data from your digital devices in a single location that you specify: It could be an encrypted server in the cloud, but it could also be a computer in a locked box under your desk. Any cellphone app, online service, or big-data research team that wants to use your data has to query your data store, which returns only as much information as is required.

**Sharing code, not data**

"The example I like to use is personalized music," says Yves-Alexandre de Montjoye, a graduate student in media arts and sciences and first author on the new paper. "Pandora, for example, comes down to this thing that they call the music genome, which contains a summary of your musical tastes. To recommend a song, all you need is the last 10 songs you listened to—just to make sure you don't keep recommending the same one again—and this music genome. You don't need the list of all the songs you've been listening to."

With openPDS, de Montjoye says, "You share code; you don't share data. Instead of you sending data to Pandora, for Pandora to define what your musical preferences are, it's Pandora sending a piece of code to you for you to define your musical preferences and send it back to them."

De Montjoye is joined on the paper by his thesis advisor, Alex "Sandy" Pentland, the Toshiba Professor of Media Arts and Sciences; Erez Shmueli, a postdoc in Pentland's group; and Samuel Wang, a software engineer at Foursquare who was a graduate student in the Department of Electrical Engineering and Computer Science when the research was done.

After an initial deployment involving 21 people who used openPDS to regulate access to their medical records, the researchers are now testing the

system with several telecommunications companies in Italy and Denmark. Although openPDS can, in principle, run on any machine of the user's choosing, in the trials, data is being stored in the cloud.

**Meaningful permissions**

One of the benefits of openPDS, de Montjoye says, is that it requires applications to specify what information they need and how it will be used. Today, he says, "when you install an application, it tells you 'this application has access to your fine-grained GPS location,' or it 'has access to your SD card.' You as a user have absolutely no way of knowing what that means. The permissions don't tell you anything."

In fact, applications frequently collect much more data than they really need. Service providers and application developers don't always know in advance what data will prove most useful, so they store as much as they can against the possibility that they may want it later. It could, for instance, turn out that for some music listeners, album cover art turns out to be a better predictor of what songs they'll like than anything captured by Pandora's music genome.

OpenPDS preserves all that potentially useful data, but in a repository controlled by the end user, not the application developer or service provider. A developer who discovers that a previously unused bit of information is useful must request access to it from the user. If the request seems unnecessarily invasive, the user can simply deny it.

Of course, a nefarious developer could try to game the system, constructing requests that elicit more information than the user intends to disclose. A navigation application might, for instance, be authorized to identify the subway stop or parking garage nearest the user. But it shouldn't need both pieces of information at once, and by requesting them, it could infer more detailed location information than the user wishes to reveal.

Creating safeguards against such information leaks will have to be done on a case-by-case, application-by-application basis, de Montjoye acknowledges, and at least initially, the full implications of some query combinations may not be obvious. But "even if it's not 100 percent safe, it's still a huge improvement over the current state," he says. "If we manage to get people to have access to most of their data, and if we can get the overall state of the art to move from anonymization to interactive systems, that would be such a huge win."

**More information:** * openpds.media.mit.edu/

* de Montjoye Y-A, Shmueli E, Wang SS, Pentland AS (2014) openPDS: Protecting the Privacy of Metadata through SafeAnswers. PLoS ONE 9(7): e98790. DOI: 10.1371/journal.pone.0098790

Provided by Massachusetts Institute of Technology