openPDS: Protecting the Privacy of Metadata through SafeAnswers

OPEN ACCESS Freely available online

PLOS ONE

openPDS: Protecting the Privacy of Metadata through SafeAnswers

Yves-Alexandre de Montjoye[1]*, Erez Shmueli[1], Samuel S. Wang[2], Alex Sandy Pentland[1]

1 Media Lab, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America, **2** DIG/CSAIL, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America

Abstract

The rise of smartphones and web services made possible the large-scale collection of personal metadata. Information about individuals' location, phone call logs, or web-searches, is collected and used intensively by organizations and big data researchers. Metadata has however yet to realize its full potential. Privacy and legal concerns, as well as the lack of technical solutions for personal metadata management is preventing metadata from being shared and reconciled under the control of the individual. This lack of access and control is furthermore fueling growing concerns, as it prevents individuals from understanding and managing the risks associated with the collection and use of their data. Our contribution is two-fold: (1) we describe openPDS, a personal metadata management framework that allows individuals to collect, store, and give fine-grained access to their metadata to third parties. It has been implemented in two field studies; (2) we introduce and analyze SafeAnswers, a new and practical way of protecting the privacy of metadata at an individual level. SafeAnswers turns a hard anonymization problem into a more tractable security one. It allows services to ask questions whose answers are calculated against the metadata instead of trying to anonymize individuals' metadata. The dimensionality of the data shared with the services is reduced from high-dimensional metadata to low-dimensional answers that are less likely to be re-identifiable and to contain sensitive information. These answers can then be directly shared individually or in aggregate. openPDS and SafeAnswers provide a new way of dynamically protecting personal metadata, thereby supporting the creation of smart data-driven services and data science research.

Citation: 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

Editor: Tobias Preis, University of Warwick, United Kingdom

Received March 14, 2014; **Accepted** May 7, 2014; **Published** July 9, 2014

Funding: This research was partially sponsored by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, by the Center for Complex Engineering Systems, and by the Media Lab Consortium. The conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* Email: yva@mit.edu

Introduction

Personal metadata – digital information about users' location, phone call logs, or web-searches – is undoubtedly the oil of modern data-intensive science [1] and of the online economy [2]. This high-dimensional metadata is what allow apps to provide smart services and personalized experiences. From Google's search to Netflix's "movies you should really watch," from Pandora to Amazon, metadata is used by commercial algorithms to help users become more connected, productive, and entertained. In science, this high-dimensional metadata is already used to quantify the impact of human mobility on malaria [3] or to study the link between social isolation and economic development [4].

Metadata has however yet to realize its full potential. This data is currently collected and stored by hundreds of different services and companies. Such fragmentation makes the metadata inaccessible to innovative services, researchers, and often even to the individual who generated it in the first place. On the one hand, the lack of access and control of individuals over their metadata is fueling growing concerns. This makes it very hard, if not impossible, for an individual to understand and manage the associated risks. On the other hand, privacy and legal concerns are preventing metadata from being reconciled and made broadly accessible, mainly because of concerns over the risk of re-identification [5–7].

Here we introduce openPDS, a field-tested personal data store (PDS) allowing users to collect, store, and give fine-grained access to their metadata to third parties. We also introduce SafeAnswers, a new and practical way of protecting the privacy of metadata through a question and answer system. Moving forward, advancements in using and mining these metadata have to evolve in parallel with considerations of control and privacy [8–11]. openPDS and SafeAnswers allow personal metadata to be safely shared and reconciled under the control of the individual.

Towards Personal Data Stores

While questions of data ownership and the creation of repositories of personal data have been discussed for a long time [12–20], their deployment on a large-scale is a chicken-and-egg problem; users are waiting for compatible services while services are waiting for user adoption. Revelations of the collection and use of metadata by governments and companies [21–22] have however recently drawn attention to their potential. The combination of 1) a public interest in questions of control but also use of their data, 2) political and legal support on data

ownership [23–26] and 3) the scale at which metadata can now be collected and processed, might trigger the large-scale deployment of PDS.

openPDS fully aligns with these trends. It uses the World Economic Forum definition of "ownership" of metadata [25]: the rights of possession, use, and disposal. It follows policies of the National Strategy for Trust Identities in Cyberspace (NSTIC) [24] and strongly aligns with the European Commission's reform of the data protection rules [23]. Finally, it recognizes that users are interacting with numerous data sources on a daily basis. Interoperability is thus not enough to achieve data ownership or address privacy concerns. Instead, openPDS implements a secure space acting as a centralized location where the user's metadata can live. openPDS can be installed on any server under the control of the individual (personal server, virtual machine, etc) or can be provided as a service (SaaS by independent software vendors or application service providers). This allows users to view and reason about their metadata and to manage fine-grained data access.

From an economic standpoint, data ownership by the individual fundamentally changes the current eco-system. It enables a fair and efficient market for metadata [2,27] – a market where users can get the best services and algorithms for their metadata. Users can decide whether a service provides enough value for the amount of data it requests, and services can be rated and evaluated. Users are empowered to ask questions like "Is finding out the name of this song worth enough to me to give away my location?" Users can seamlessly give new services access to their past and present metadata while retaining ownership. From a business standpoint, such data ownership is likely to help foster alternatives to the current data-selling and advertising-based business model. New business models focusing on providing hardware for data collection, storage for metadata, or algorithms for better using metadata might emerge while software for data collection and data management might be mostly open-source. The proposed framework removes barriers to entry for new businesses, allowing the most innovative algorithmic companies to provide better data-powered services [2].

Other approaches have been proposed for the storage, access control, and privacy of data. Previous approaches fall into two categories: cloud storage systems and personal data repositories. First, cloud storage systems, such as the ones that have been commercially developed by companies like Dropbox [28] and Carbonite [29], are a first approximation of a user-controlled information repository for personal data. They however focus on storing files and only implement the most basic type of access control, usually on a file or folder basis. They do not suggest any data aggregation mechanisms and, once access has been granted, the raw data is exposed to the outer world, potentially compromising privacy. Second, personal data repositories have been developed in academic [12–17,30,31] and commercial settings [18–20]. All of these repositories are however restricted to specific queries on a particular type of data, such as interests or social security numbers. They provide only a basic access-control level, which means that once access to the data is authorized, privacy may be compromised. openPDS differs from previous approaches in its alignment with current political and legal thinking, its focus on large-scale metadata, and its SafeAnswers privacy-preserving mechanism.

On Privacy

There is little doubt that web searches, GPS locations, and phone call logs contain sensitive private information about an individual. In 2012, 72 percent of Europeans were already concerned about the use of their personal data [23]. The recent revelations are unlikely to have helped [21,22]. Addressing users' legitimate privacy concerns will soon be a prerequisite to any metadata usage.

Protecting the privacy of metadata is known to be a hard problem. The risks associated with high-dimensional metadata are often subtle and hard to predict and anonymizing them is known to be very hard. Over the last years, numerous works have exposed the risks of re-identification or de-anonymization of apparently anonymous datasets of metadata. An anonymous medical database was combined with a voters' list to extract the health record of the governor of Massachusetts [7] while the Kinsey Institute database was showed to be re-identifiable using demographics [32]. Twenty million web queries from around 650,000 AOL users were found to be potentially re-identifiable thanks to people's vanity searches [33] while the Netflix challenge dataset was de-anonymized using users' ratings on IMDB (The Internet Movie Database) [6]. Finally, mobility datasets of millions of users were found to be potentially re-identifiable using only four approximate spatio-temporal points [5].

Geospatial metadata, the second most recorded information by smartphone applications [34,35], is probably the best example of the risks and rewards associated with metadata [36]. On the one hand, a recent report of the Electronic Frontier Foundation [37] worries about potentially sensitive information that can be derived from geospatial metadata. For example, geo-spatial metadata behavior collected from mobile phones has been shown to be very useful in predicting users' personalities [38]. On the other hand, the number of users of location-aware services, such as Yelp or Foursquare, are rising quickly as these services demonstrate their benefits to users.

Numerous ways of anonymizing personal data beyond the simple removal of explicit identifiers have been proposed. Similar to the original k-anonymity model [39], they aim minimize privacy risks while keeping data utility as high as possible. All anonymization models have however several major limitations.

Generic anonymization models have been designed for relatively low-resolution data and cannot be easily extended to high-dimensional data such as GPS location or accelerometer readings. Through generalization and suppression, k-anonymity makes every record in a given table indistinguishable from at least $k-1$ other records, thereby making it impossible to identify an individual in that table. Variations and alternatives include ℓ-diversity [40], which address attacks based on lack of diversity in the sensitive data and t-closeness [41,42] which aims at maintaining the distribution of the sensitive data. The reader is referred to the surveys [43,44] for further details. In metadata, any information that is unique to an individual can be used to re-identify him. Unicity ($\mathcal{E}$) has been used to quantify the re-identifiability of a dataset [5]. Most rich metadata datasets are expected to have a high $\mathcal{E}$. This means that, even if they are computationally tractable, generic privacy models are likely to result in most data having to be suppressed or generalized to the top-most values in order to satisfy the privacy requirement [45]. This curse of dimensionality led to the development of models dedicated to the anonymization of mobility data.

Mobility-focused anonymization models protect individual's mobility traces but only for very specific data applications or against specific re-identification attacks. The anonymization models in [46–50] protect the current location of the user, allowing him to anonymously perform accurate location-based searches. They however prevent any uses of historical metadata or side information, making them impractical for research and smart services using historical data. Other models [51,52] allow for the anonymization of short successions of geospatial locations with no

associated timestamps or [53] protect an individual's mobility data against re-identification at certain given times. These models however focus on anonymizing mobility data with a certain purpose or specific type of data in mind (i.e., current location, trajectory without timestamps or mobility data in given times). This makes these models impracticable for most data-science applications in academia and organizations.

Finally, all anonymization models, generic or mobility-focused, assume a setting in which the whole database is anonymized and published once. This makes it impractical, as (1) the same database is likely to be used to address different research questions (which might need specific pieces of information) and (2) smartphone applications or researchers might need access to the very latest pieces of information. Modifying existing anonymization models to support multiple releases has been shown to be non-trivial [54]. Indeed, anonymizing each publication on its own is not sufficient, since a violation of privacy may emerge as a result of joining information from different publications. Anonymizing the whole database once and successively releasing the relevant part of the anonymized data is not a solution either, since newer data may become available. Several dedicated models were recently suggested to address the multiple publications setting [54–57]. While very interesting, these models are based on extensions of the original one publication models and are thus very limited in the number and type of publications that they can handle.

SafeAnswers, a new paradigm

The goal of SafeAnswers is to turn an algorithmically hard anonymization and application-specific problem into a more tractable security one by answering questions instead of releasing copies of anonymized metadata.

Under the openPDS/SafeAnswers mechanism, a piece of code would be installed inside the user's PDS. The installed code would use the sensitive raw metadata (such as raw accelerometers readings or GPS coordinates) to compute the relevant piece of information within the safe environment of the PDS. In practice, researchers and applications submit code (the question) to be run against the metadata, and only the result (the answer) is sent back to them. openPDS/SafeAnswers is similar to differential privacy [58,59], both being online privacy-preserving systems. Differential Privacy is however designed for a centralized setting where a database contains metadata about numerous individuals and answers are aggregate across these individuals. SafeAnswers is unique, as it focuses on protecting the privacy of a single individual whose data are stored in one place by reducing the dimensionality of the metadata before it leaves the safe environment. This individual-centric setting makes it practical for mobile applications or data-science researchers. It however introduces new privacy challenges [see Analysis].

Combined with openPDS, this simple idea allows individuals to fully use their data without having to share the raw data. SafeAnswers also allows users to safely grant and revoke data access, to share data anonymously without needing a trusted third-party, and to monitor and audit data uses [Fig. 1 and 2].

Results

The openPDS framework

The Dataflow. Looking at Fig. 3, consider a usecase in which a user uses a personalized music service such as PersonalizedMusic. Every time PersonalizedMusic needs to decide which song to play next on the user's mobile phone or desktop, it sends a request to the user's PDS. The actual computation of what song to play next is done by the PersonalizedMusic SafeAnswers module (SA module) inside the PDS front-end. As part of this processing, the PersonalizedMusic SA module accesses the back-end database in order to retrieve the required metadata. The PersonalizedMusic SA module would only access the raw metadata that it was authorized to when it was installed and all the processing would take place in a software sandbox. Upon completing its processing, the PersonalizedMusic SA module would return the name of the next song to play back to the front-end who will validate it and send it back to PersonalizedMusic.

The Database. Metadata are currently stored in a CouchDB database. CouchDB is a NoSQL store that stores data as a key to document mapping, where documents are JSON objects. CouchDB also provides a large range of existing functionality that lends itself well to the type of analysis needed to compute answers or reduce the dimensionality of the metadata. It has built-in support for MapReduce through CouchDB-Views, as well as data validation. All SafeAnswers modules share one unified database, and each SA module has a corresponding key prefix.

The Front-End. The front-end ensures that no unauthorized operations are carried out on the underlying metadata. SA modules are restricted to reading from the data sources they have explicitly listed as dependencies. CouchDB can also enforce access based on metadata types, time of access, time of collection, etc. The access control mechanism is implemented based on Django users and a permissioning system, where each app is registered as a user. We are working to decouple the access control mechanism and the PDS using oAuth1.0 protocol [60]. This will allow an authentication server to hand out tokens associated with a specific service and set of metadata. In addition, SA modules are executed in a sandboxed environment, and all communications are encrypted using 256 bits SSL connections. In some implementations, PDSs can be managed from a web interface.

SafeAnswers is one key innovation of the openPDS framework. SafeAnswers allows for computations on user metadata to be performed within the safe environment of the PDS. Only safe answers, the exact information needed to provide the service, leave the PDS. SA modules are intimately tied to the notion of Design Documents in CouchDB. A CouchDB design document is intended to be a document that describes an application to be built on top of an underlying CouchDB instance. Each access of the SA module to the database has to be authorized and each SA module executes inside a sandbox. We are now working to add additional fields to the CouchDB design document specification to allow additional functionality, like SA module dependencies and permissions. These descriptions will be written in the SA module manifest to be programmatically enforced and to be presented to the user before installation.

In large-scale deployments, we expect that, instead of developing a SA module from scratch for each app, there will be common libraries that can be leveraged by SA modules or directly through a standard API. For example, there could be a library that supports functionality, like returning the current city a user is in [15], his radius of gyration in the past 7 days [61] or whether he is currently running. In the future, we also hope to further develop the SafeAnswers system to support sessions. This would allow for some of the most advanced data-science uses.

Field-studies and user feedback

Our two initial deployments offer a first qualitative evaluation of the system. The first field study is monitoring the daily behavior of individuals with diagnosed mental problems (PTSD, depression) and controls subjects for a month through their smartphones [62]. Data is used to reproduce the diagnoses of mental health conditions, focusing on changes in speech and social behavior.

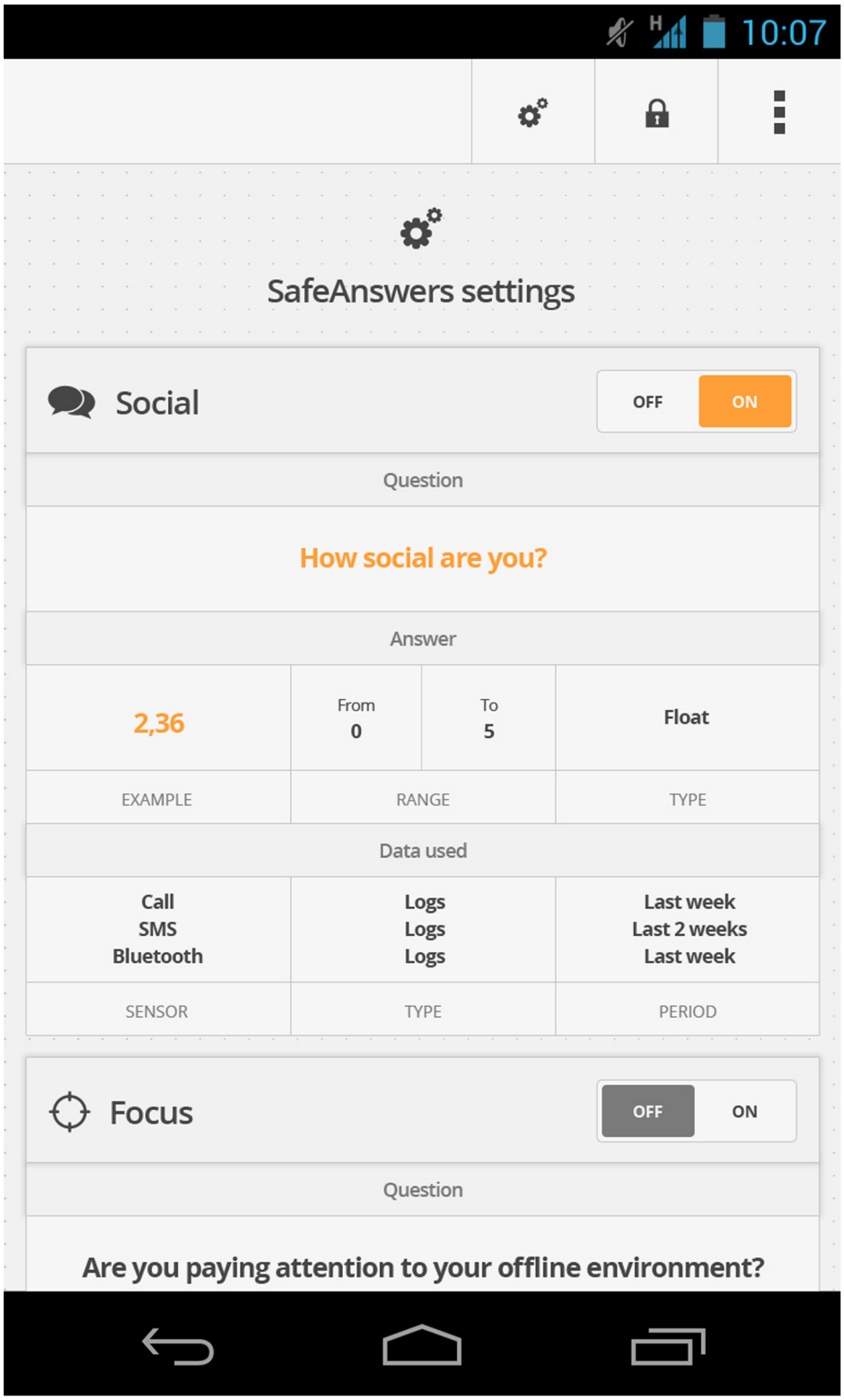

Figure 1. Mockups of the proposed SafeAnswers settings presented to the user for approval. This screen shows the question answered, examples of the possible responses, and the sensors used to compute the response.
doi:10.1371/journal.pone.0098790.g001

Recorded activities include psycho-motor activity, occupational activity, social interaction, and sleep behavior.

Fig. 4 presents "focus-group" results about the reaction of individuals to the openPDS framework ($N = 21$, 6 females and 15 males, median age category is 29 to 34 old). We consider the

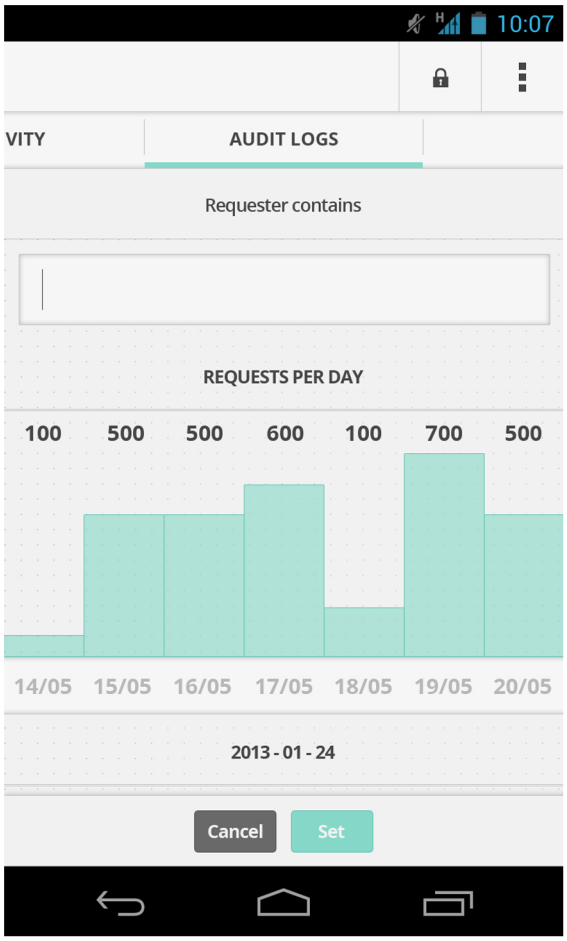

Figure 2. Mockups of the proposed interface showing the number of requests sent by a given app per day.
doi:10.1371/journal.pone.0098790.g002

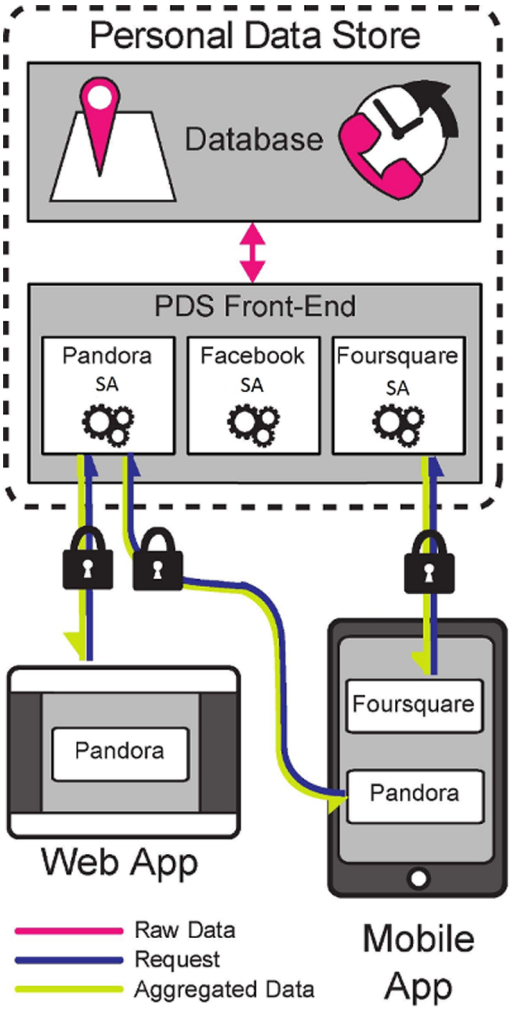

Figure 3. openPDS system's architecture. LBSinc web or mobile app sent a request to the user's openPDS. The request is passed on to the LBSinc SA module, which requests access to the database in order to retrieve the metadata needed to compute the answer. The SA module computes the answer, which is then validated by the PDS Front-End and send back to the web or the mobile app.
doi:10.1371/journal.pone.0098790.g003

deployment to be a success, as 81% of individuals say they would use it in their personal life and, on a 1 to 5 scale (1: "Not at all comfortable" and 5: "Extremely comfortable"), are comfortable with the data collection (mean: 4, sem: 0.27). From a privacy perspective, we can see that the ability to delete data matters to participants (mean: 4.10, sem: 0.27). We can qualitatively see that users are generally comfortable sharing individual data with their primary care provider and mental health specialist. However, they seem to be less comfortable sharing such data with friends and potentially their family members. We can also see that anonymity matters to participants (mean:4 sem:0.30) and that they are significantly more comfortable sharing anonymous, rather than individual, data (p-value <0.005 with a one-tailed, paired, non-parametric Kolmogorov-Smirnov test on 4 specific sharing questions, and mean:4 sem:0.25 when asked on the importance of anonymizing shared data). All these emphasize the relevance of the openPDS/SafeAnswers framework.

A second study, the mobile territorial lab, in partnership with Telecom Italia, Telefonica, and the Fondazione Bruno Kessler, is now underway. It is composed of 70 young parents living in Trento and its premises. The aim here is to create a long-term living lab to study user behavior and to perform user studies. Participants' behavior is recorded using an extended version of the open-sensing framework FunF [63]. All collected metadata are stored on users' PDSs.

Discussion

Performance

openPDS may introduce a performance overhead caused by its distributed nature, the added security and privacy mechanisms and the group computation mechanism [see Analysis].

First, the distributed nature of openPDS requires services to access the user's PDS when an answer has to be computed. In cases where computing the answer is fast, the latency it imposes might make an openPDS-based solution impracticable. Solutions such as precomputing some values and locally caching them might help. However, in cases where computing the answer inside the PDS dominates the total execution time, this might not significantly impact the user experience. In fact, this might actually introduce a performance boost, since it parallelizes the computations that are being performed at a per-PDS level.

Second, the added security and privacy mechanisms described below may also result in performance overhead. This overhead needs to be taken into account when choosing the appropriate mechanism. For example, the on-the-fly nature of openPDS/

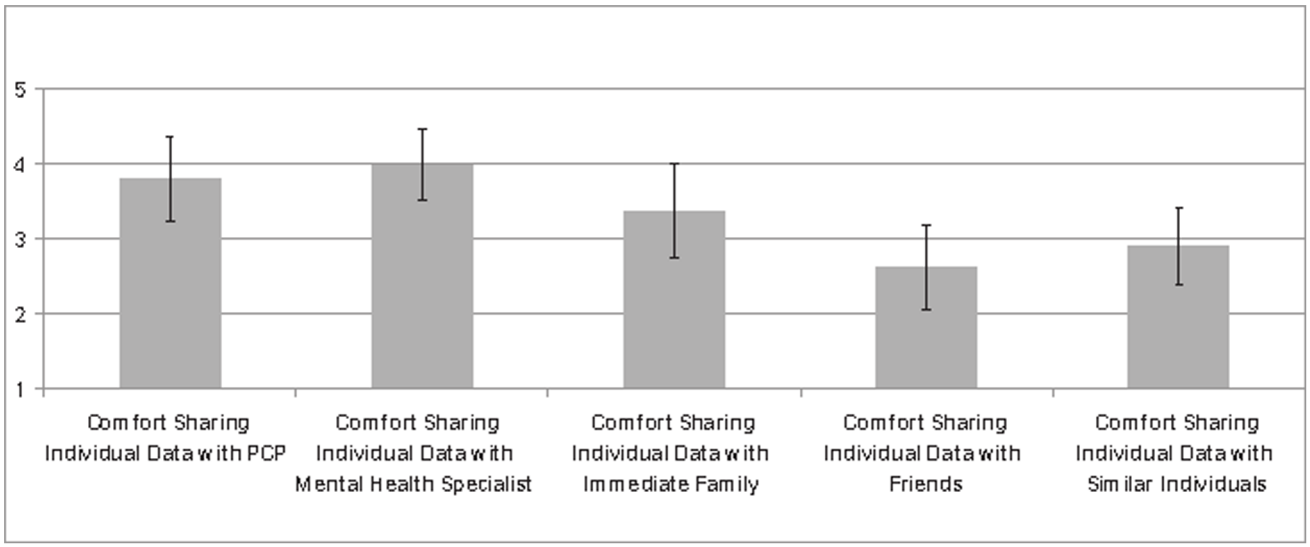

Figure 4. Individuals' reaction to data sharing. The error bars are bootstrapped 95% high-density intervals. We can qualitatively see that users are generally comfortable sharing individual data with their primary care provider and mental health specialist. They however seem to be less comfortable sharing such data with friends and potentially family members.
doi:10.1371/journal.pone.0098790.g004

SafeAnswers may lead to inference of sensitive data if the results of several queries are joined together. On the one hand, using techniques such as the one suggested by [54] may be very efficient in preventing such inference, but they are relatively expensive in computation time. On the other hand, adding noise to query results may not be equally efficient, but would result in a much faster computation time. Advanced techniques might thus be crucial when dealing with credit card or location data, but noise addition might be sufficient to protect less sensitive data such as accelerometer readings.

For many years, group computation has been of theoretical interest only. Great improvements and actual field-studies in domains such as electronic voting, actioning, and data mining have recently made group computation–also called Secure Multiparty Computation, or SMC–of practical interest [64]. Similar to network latency, the overhead of SMC might become reasonable if computing the answer dominates the total computation time. SMC has furthermore recently been generalized into belief propagation algorithms [65]. This means that every node of the computation does not have to communicate with every other anymore, thereby reducing the overhead.

Usage Experience

In this section we describe two short scenarios for a user and a developer switching to an openPDS/SafeAnswers system for mobile applications.

End-User. Suppose Alice wants to install and use a smartphone app like LBSinc, a location-based check-in application, without using a PDS. Alice downloads the app onto her phone, authorizes LBSinc to access her phone's network communication and GPS coordinates, and creates a user account with LBSinc. The LBSinc app starts collecting metadata about her and stores it all in its back-end servers. Under this model it is difficult for Alice to access the metadata LBSinc uses to makes inferences about her, or to remove the metadata she does not want LBSinc to access or use.

Alternatively, Alice could decide to download a PDS-aware version of LBSinc. She installs it just like she would install any other smartphone app and authorizes it to access only her phone's network communication. When used for the first time, the smartphone app prompts her to enter her PDS URI. Alice then sees exactly what metadata the LBSinc SA module will have access to and examples of the answers [see Fig. 2], the relevant summarized information that will be send back to LBSinc. If she accepts, the LBSinc SA module is installed onto her PDS and she can start using it.

App Developer. Suppose a developer now wants to implement MyMusic, a smartphone app that plays music to Alice based on her preferences and current activity. Under the current model, he would first have to develop a smartphone app to collect the metadata on Alice's phone, record it, and periodically send it to a server. He would then develop a server with an internal database to store the raw activity data he collects, a secured API for this database to receive the metadata, and a way to anonymize the metadata or at least separate the user account information from the metadata. He could then start developing an algorithm to decide which song or type of music to play. The initial picture he would have of users would be very rough, as he would have no prior metadata to work with. Finally, he would have to wait to collect a sufficient amount of metadata before being able to provide adequate recommendations.

If operating within the openPDS/SafeAnswers framework, the metadata that the developer needs are likely to have already been collected either by a metadata collection app [66] or by another application or service. The developer would then spend most of his time writing an SA module that would decide which song or type of music to play and test it on development copies of PDSs. The PDS front-end would take care of creating the API and of securing the connection for him. The developer's algorithm would be able to access a potentially large set of metadata, including historical metadata.

Analysis

The openPDS framework suggests several mechanisms for enhancing the privacy and security of personal metadata: SafeAnswers, access control, sandboxes, and network encryption. In

this section, we discuss several cases where these might fall short and discuss potential counter-measures.

Protecting aggregated answers of groups

A practical example would be a service, such as CouponInc, which wants to execute a simple query to know how many of its users are around a certain shop to send them a special coupon. CouponInc might want to issue a query like "How many users are in this geographical area at the current time?" or "How active are these users during lunch time?"

In a distributed setting, such computation falls under the well-studied field of secure multi-party computation (SMC) [67], where the querying agent never sees any individual user's metadata but can access information aggregated across users. User privacy is preserved, as each PDS only sends cryptographically masked messages to other nodes in the network.

Such a cryptographic technique fits elegantly into the PDS model of computation [Fig. 5]. Rather than anonymizing and computing over-complex data items, like GPS coordinates, the SA modules could compute features locally to each user's PDS, reducing the dimensionality of the metadata. After the local computation is done, the inferred facts–e.g. whether or not a given user is in a given geographical area–can be aggregated in a privacy-preserving way. This means that even the low-dimension answer cannot be associated with a particular user.

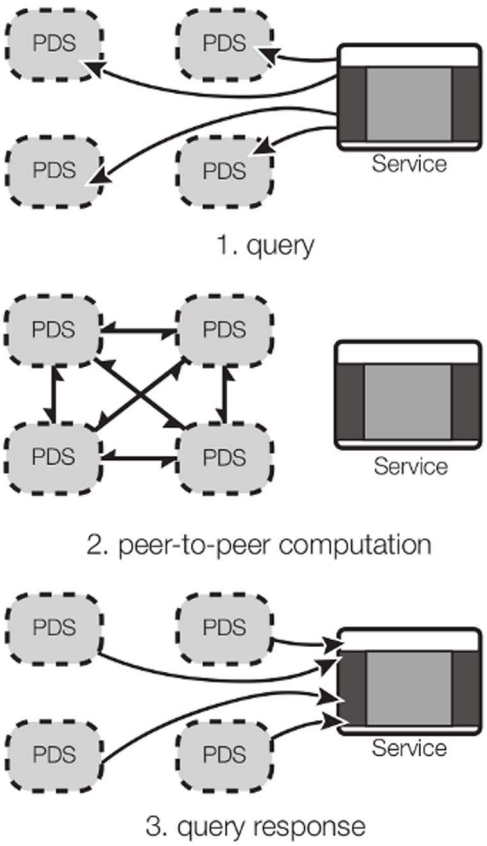

Figure 5. Group Computation Overview. (1) A querying agent (like CouponInc) passes a function that its wants a collaborative answer for, along with a list of URI to PDSs. (2) PDSs all trade messages in order to compute a collaborative answer. (3) The answer is reported back to the querying agent.
doi:10.1371/journal.pone.0098790.g005

Attacks in the case of well-behaved apps

Even in the absence of attackers, apps that behave as they are supposed to might pose a risk to users' privacy. We notice two major challenges: (1) How can an openPDS/SafeAnswers determine the required level of aggregation given that it only has access to the metadata of a single user? (2) Well-behaved apps could inadvertently collect data whose combinations may allow others to infer sensitive information.

A potential solution to the first challenge might be found in [5]. The authors studied fifteen months of human mobility data for one and a half million individuals, and found that one formula determines the uniqueness of an individual's mobility traces, given the traces' resolution (i.e., level of aggregation) and the amount of background knowledge available to the adversary. If extended to other types of data, such an equation could be used by SafeAnswers to determine the required level of aggregation needed when answering a query.

The fields of Privacy Preserving Data Publishing and Mining aim to address a problem similar to the second challenge: how to anonymize the current publication of a database so that the combination of all past and current anonymized publications respect privacy. These works suggest several interesting assumptions and techniques that could be adopted by the openPDS/SafeAnswers framework. For example, the authors of [54] show that the problem of accurately calculating the level of privacy imposed by a set of three or more publications is NP-hard. The authors then suggest a relaxed method for calculating the privacy level in polynomial time. Their method is based on joining the set of publications into a single table, which can then be checked against some privacy requirement. They also suggest a supplementing algorithm for anonymizing the current publication so that the required privacy level is obtained. Their methods might be used by SafeAnswers in order to determine whether the current set of queries and potential future queries might compromise privacy.

Work in statistical databases might also help address the second challenge [68]. A statistical database aims to allow the execution of statistical queries without compromising the confidentiality of any individual represented in the database. Two approaches used in this field could be useful for SafeAnswers: (1) A query restriction rejects each query that could compromise a user's privacy and provides accurate answers to legitimate queries. The computation of what is a legitimate query is usually based on the size of the query's results or the extent of overlap between queries. Note however that the denial of a query may, in itself, provide information to an an attacker. (2) Perturbation gives approximate answers by adding noise to the answers computed from the original metadata. Regardless of the specific perturbation technique, the designer must attempt to produce statistics that accurately reflect the underlying database. Such perturbed answers might however not be acceptable for all uses.

Attacks in the case of malicious apps

While well-behaved apps might inadvertently collect sensitive information, apps that are voluntarily not playing by the rules pose a serious threat to user's privacy. The major risk we see here is how to protect the metadata against an app that deliberately tries to infer sensitive information by over-querying a user's openPDS or by colluding with other apps.

Technically, numerous techniques from anomaly detection may help SafeAnswers detect suspicious behavior. For example, a service that suddenly changes its query pattern; querying for location every minute while it used to ask user's location and speed a few times in a row 3 times a day. The detection of anomalies, outliers, or rare events, has recently gained a lot of attention in

many security domains, ranging from video surveillance and security systems to intrusion detection and fraudulent transactions. Accordingly [69], most anomaly detection methods are based on the following techniques: classification, nearest neighbor, clustering, statistical, information theoretic, and spectral. Any of these techniques, or their combination, can potentially be used by SafeAnswers.

Anomaly detection could also be combined with reputation systems to allow for a group of openPDSs to exchange information about modules and services in real-time. The P2P reputation systems literature considers different types of malicious behavior that can be blocked with the help of reputation systems. These give us a foretaste of potential risks. "Traitors" are services who initially behave properly but then start to misbehave and inflict damage on the community. This technique is particularly effective when the service has become respectable and well installed. "Whitewashers" are services who leave and rejoin the system with new identities in order to purge the bad reputations they acquired under their previous identities. Finally, "Collusions" are a group of malicious services acting together to cause damage. Such reputation systems could be combined with other privacy mechanisms discussed here. For example, an openPDS might decide to allow a service with a medium rating to execute restricted or noisy queries but temporarily block a service whose rating suddenly dropped.

Various UI mechanisms can also be used to warn users of potentially malicious apps before they are installed. For example, trust could be used to rate service providers. Adapting the definition from [70], trust would reflects a user's or a PDS's subjective view of a service, while reputation could be considered a collective measure of trust reflecting a group view of that service. Work by [71] shows that the reputation of the service provider matters more than the specific data being accessed and hints at the potential usefulness of a reputation system to help users decide which services to trust. Various principles for computing reputation and trust can be found in [72]. Besides a simple summation or average of ratings, the authors mention discrete models in which trust is a discrete value from a predefined set of values, fuzzy models, bayesian systems, belief models, and flow models.

Attacks compromising the host

Finally, openPDS is vulnerable to the traditional security and privacy issues of any hosted system. Attackers could compromise the authentication/control mechanisms or impersonate existing users to gain access to the database or to corrupt the system. For instance, in the case of virtual machines hosting openPDSs, an attacker's virtual machine can legitimately be located in the same physical machine as openPDSs virtual machines. This is, however, not specific to openPDS, and similar issues exist with any hosted systems, such as SaaS, virtual machine and traditional servers. Solutions include hypervisors [73] or data-at-rest encryption [74,75] such as homomorphic encryption schemes [76]. The main difference openPDS introduces is having the data distributed across machines, systems, and implementations of openPDS. While a full analysis is beyond the scope of this paper, one might imagine that a distributed and heterogeneous system might be harder to attack than some of the traditional centralized ones especially if information is shared across machines [see previous section].

Conclusion

Finally, as technologists and scientists, we are convinced that there is an amazing potential in personal metadata, but also that benefits should be balanced with risks. By reducing the dimensionality of the metadata on-the-fly and actively protecting users, openPDS/SafeAnswers opens up a new way for individuals to regain control over their privacy.

openPDS/SafeAnswers however still face a number of challenges. Each challenges includes several potential directions for future research: (1) the automatic or semi-automatic validation of the processing done by a PDS module; (2) the development of SafeAnswers privacy-preserving techniques at an individual level for high-dimensional and ever-evolving data (mobility data, accelerometer readings, etc.) based on existing anomaly detection framework and potentially stored in highly-decentralized systems; (3) the development or adaptation of privacy preserving data-mining algorithms to an ecosystem consisting of distributed PDSs; and (4) UIs allowing the user to better understand the risks associated with large-scale metadata and to monitor and visualize the metadata used by applications.

Acknowledgments

The authors would like to thank Hal Abelson, Fabrizio Antonelli, John Clippinger, Alan Gardner, Dazza Greenwood, Bruno Lepri, Wei Pan, Henrik Sandell, Jeff Schmitz, Brian Sweatt, Michele Vescovi, and the ID^3 foundation for helpful conversations; Skyler Place for sharing data; and Danielle Hicks, Jeff Schmitz, and Cody Sumter for help with the design.

Author Contributions

Analyzed the data: YdM ES. Contributed reagents/materials/analysis tools: YdM ES SW. Wrote the paper: YdM ES SW AP.

References

1. Lazer D, Pentland AS, Adamic L, Aral S, Barabasi AL, et al. (2009) Life in the network: the coming age of computational social science. Science (New York, NY) 323: 721.
2. Schwab K, Marcus A, Oyola J, Hoffman W, Luzi M (2011) Personal data: The emergence of a new asset class. In: An Initiative of the World Economic Forum.
3. Wesolowski A, Eagle N, Tatem AJ, Smith DL, Noor AM, et al. (2012) Quantifying the impact of human mobility on malaria. Science 338: 267–270.
4. Eagle N, Macy M, Claxton R (2010) Network diversity and economic development. Science 328: 1029–1031.
5. de Montjoye YA, Hidalgo CA, Verleysen M, Blondel VD (2013) Unique in the crowd: The privacy bounds of human mobility. Nature SRep 3.
6. Narayanan A, Shmatikov V (2008) Robust de-anonymization of large sparse datasets. In: Security and Privacy, 2008. SP 2008. IEEE Symposium on. IEEE, pp. 111–125.
7. Sweeney L (2002) k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10: 557–570.
8. de Montjoye YA, Wang SS, Pentland A, Anh DTT, Datta A, et al. (2012) On the trusted use of large-scale personal data. IEEE Data Eng Bull 35: 5–8.
9. Palfrey J, Zittrain J (2011) Better data for a better internet. Science 334: 1210–1211.
10. Abelson H, Ledeen K, Lewis H (2008) Blown to bits: your life, liberty, and happiness after the digital explosion. Addison-Wesley Professional.
11. Rubinstein IS (2012) Big data: The end of privacy or a new beginning?
12. Bell G (2001) A personal digital store. Communications of the ACM 44: 86–91.
13. Want R, Pering T, Danneels G, Kumar M, Sundar M, et al. (2002) The personal server: Changing the way we think about ubiquitous computing. Ubicomp 2002: Ubiquitous Computing: 223–230.
14. Baden R, Bender A, Spring N, Bhattacharjee B, Starin D (2009) Persona: an online social network with user-defined privacy. In: ACM SIGCOMM Computer Communication Review. ACM, volume 39, pp. 135–146.
15. Mun M, Hao S, Mishra N, Shilton K, Burke J, et al. (2010) Personal data vaults: a locus of control for personal data streams. In: Proceedings of the 6th International Conference. ACM, p. 17.
16. Cáceres R, Cox L, Lim H, Shakimov A, Varshavsky A (2009) Virtual individual servers as privacypreserving proxies for mobile devices. In: Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds. ACM, pp. 37–42.

17. Hong J, Landay J (2004) An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the 2nd international conference on Mobile systems, applications, and services. ACM, pp. 177–189.
18. Higgins website. URL http://www.eclipse.org/higgins/. Accessed 2014 May.
19. Mydex website. URL http://mydex.org/. Accessed 2014 May.
20. Azigo website. URL https://www.azigo.com/. Accessed 2014 May.
21. Gellman B, Soltani A (2013) NSA tracking cellphone locations worldwide, snowden documents show. The Washington Post.
22. Greenwald G, MacAskill E (2013) NSA prism program taps in to user data of apple, google and others. The Guardian.
23. European Commission proposes a comprehensive reform of data protection rules to increase users' control of their data and to cut costs for businesses. URL http://europa.eu/rapid/pressReleasesAction.do?reference = IP/12/46 &format = HTML&aged = 0&language = EN&guiLanguage = en. Accessed 2014 May.
24. National Strategy for Trust Identities in Cyberspace. URL http://www.whitehouse.gov/sites/default/files/rss_viewer/NSTICstrategy_041511.pdf. Accessed 2014 May.
25. Reality Mining of Mobile Communications: Toward a New Deal on Data. URL https://members.weforum.org/pdf/gitr/2009/gitr09fullreport.pdf. Accessed 2014 May.
26. International Strategy for Cyberspace. URL http://www.whitehouse.gov/sites/default/files/rss_viewer/internationalstrategy_cyberspace.pdf. Accessed 2014 May.
27. Schwartz P (2003) Property, privacy, and personal data. Harv L Rev 117: 2056.
28. Dropbox website. URL http://www.dropbox.com/. Accessed 2014 May.
29. Carbonite Backup website. URL http://www.carbonite.com/en/. Accessed 2014 May.
30. Kay J, Kummerfeld B (2012) Creating personalized systems that people can scrutinize and control: Drivers, principles and experience. ACM Transactions on Interactive Intelligent Systems (TiiS) 2: 24.
31. Assad M, Carmichael DJ, Kay J, Kummerfeld B (2007) Personisad: Distributed, active, scrutable model framework for context-aware services. In: Pervasive Computing, Springer. pp. 55–72.
32. Solomon A, Hill R, Janssen E, Sanders SA, Heiman JR (2012) Uniqueness and how it impacts privacy in health-related social science datasets. In: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium. ACM, pp. 523–532.
33. Butler D (2007) Data sharing threatens privacy. Nature 449: 644.
34. Thurm S, Kane YI (2014) Your Apps Are Watching You. The Wall Street Journal.
35. The App Genome Project Lookout website. URL http://blog.myLookout.com/. Accessed 2014 May.
36. Stopczynski A, Pietri R, Pentland A, Lazer D, Lehmann S (2014) Privacy in sensor-driven human data collection: A guide for practitioners. arXiv preprint arXiv:14035299.
37. Blumberg A, Eckersley P (2009) On Locational Privacy and how to avoid losing it forever. EFF.
38. de Montjoye YA, Quoidbach J, Robic F, Pentland AS (2013) Predicting personality using novel mobile phone-based metrics. In: Social Computing, Behavioral-Cultural Modeling and Prediction, Springer. pp. 48–55.
39. Sweeney L (2002) *k*-Anonymity: A model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10: 557–570.
40. Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M (2006) *l*-Diversity: privacy beyond *k*-anonymity. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06). p. 24.
41. Cao J, Karras P, Kalnis P, Tan K (2011) Sabre: a sensitive attribute bucketization and redistribution framework for *t*-closeness. The VLDB Journal 20: 59–81.
42. Li N, Li T, Venkatasubramanian S (2010) Closeness: A new privacy measure for data publishing. IEEE Transactions on Knowledge and Data Engineering 22: 943–956.
43. Aggarwal C, Yu P (2008) Privacy-preserving data mining: models and algorithms. Springer-Verlag New York Inc.
44. Fung B, Wang K, Chen R, Yu P (2010) Privacy-preserving data publishing: a survey of recent developments. ACM Computing Surveys (CSUR) 42: 1–53.
45. Aggarwal C (2005) On k-anonymity and the curse of dimensionality. In: Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, pp. 901–909.
46. Beresford A, Stajano F (2003) Location privacy in pervasive computing. Pervasive Computing, IEEE 2: 46–55.
47. Gedik B, Liu L (2005) Location privacy in mobile systems: A personalized anonymization model. In: Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on. Ieee, pp. 620–629.
48. Zhong G, Goldberg I, Hengartner U (2007) Louis, lester and pierre: Three protocols for location privacy. In: Proceedings of the 7th international conference on Privacy enhancing technologies. Springer-Verlag, pp. 62–76.
49. Mascetti S, Freni D, Bettini C, Wang X, Jajodia S (2011) Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. The VLDB JournalThe International Journal on Very Large Data Bases 20: 541–566.
50. Reades J (2010) Finite state machines: preserving privacy when data-mining cellular phone networks. Journal of Urban Technology 17: 29–40.
51. Monreale A, Andrienko G, Andrienko N, Giannotti F, Pedreschi D, et al. (2010) Movement data anonymity through generalization. Transactions on Data Privacy 3: 91–121.
52. Terrovitis M, Mamoulis N, Kalnis P (2008) Privacy-preserving anonymization of set-valued data. Proceedings of the VLDB Endowment 1: 115–125.
53. Yarovoy R, Bonchi F, Lakshmanan L, Wang W (2009) Anonymizing moving objects: how to hide a mob in a crowd? In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. ACM, pp. 72–83.
54. Shmueli E, Tassa T, Wasserstein R, Shapira B, Rokach L (2012) Limiting disclosure of sensitive data in sequential releases of databases. Information Sciences 191: 98–127.
55. Wang K, Fung B (2006) Anonymizing sequential release. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06). pp. 414–423.
56. Byun JW, Sohn Y, Bertino E, Li N (2006) Secure anonymization for incremental datasets. In: Secure Data Management. pp. 48–63.
57. Xiao X, Tao Y (2007) M-invariance: towards privacy preserving re-publication of dynamic datasets. In: Proceedings of the 2007 ACM SIGMOD international conference on Management of Data (SIGMOD '07). pp. 689–700.
58. Dwork C (2006) Differential privacy. In: Automata, languages and programming, Springer. pp. 1–12.
59. Mir DJ, Isaacman S, Cáceres R, Martonosi M, Wright RN (2013) Dp-where: Differentially private modeling of human mobility. In: Big Data, 2013 IEEE International Conference on. IEEE, pp. 580–588.
60. Hammer-Lahav E (2010) Rfc 5849: The oauth 1.0 protocol. Internet Engineering Task Force (IETF).
61. Gonzalez MC, Hidalgo CA, Barabasi AL (2008) Understanding individual human mobility patterns. Nature 453: 779–782.
62. Place RCAA S, Feast J (March, 2013) Evaluation of trust framework sharing and privacy concerns. In: Technical Report 030113. Cogito Corporation.
63. FunF website. URL http://www.funf.org/. Accessed 2014 May.
64. Orlandi C (2011) Is multiparty computation any good in practice? In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, pp. 5848–5851.
65. Kearns M, Tan J,Wortman J (2007) Privacy-preserving belief propagation and sampling. Advances in Neural Information Processing Systems 20.
66. Behav.io website. URL http://behav.io/. Accessed 2014 May.
67. Goldreich O (1998) Secure multi-party computation. Manuscript Preliminary version.
68. William S (2008) Computer Security: Principles And Practice. Pearson Education India.
69. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: A survey. ACM Computing Surveys (CSUR) 41: 15.
70. Mui L, Mohtashemi M, Halberstadt A (2002) A computational model of trust and reputation. In: System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on. IEEE, pp. 2431–2439.
71. Lederer S, Mankoff J, Dey A (2003) Who wants to know what when? privacy preference determinants in ubiquitous computing. In: CHI'03 extended abstracts on Human factors in computing systems. ACM, pp. 724–725.
72. Jøsang A, Ismail R, Boyd C (2007) A survey of trust and reputation systems for online service provision. Decision support systems 43: 618–644.
73. Popa L, Yu M, Ko S, Ratnasamy S, Stoica I (2010) Cloudpolice: taking access control out of the network. In: Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks. ACM, p. 7.
74. Shmueli E, Vaisenberg R, Gudes E, Elovici Y (2014) Implementing a database encryption solution, design and implementation issues. Computers & Security.
75. Popa RA, Redfield C, Zeldovich N, Balakrishnan H (2011) Cryptdb: protecting confidentiality with encrypted query processing. In: Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. ACM, pp. 85–100.
76. Gentry C (2009) A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University.