# SensorSafe: a Framework for Privacy-Preserving Management of Personal Sensory Information

Haksoo Choi, Supriyo Chakraborty, Zainul M. Charbiwala, and Mani B. Srivastava

University of California, Los Angeles
{haksoo,supriyo,zainul,mbs}@ucla.edu

**Abstract.** The widespread use of smartphones and body-worn sensors has made continuous and unobtrusive collection of personal data feasible. This has led to the emergence of useful applications in diverse areas such as medical behavioral studies, personal health-care and participatory sensing. However, the nature of highly personal information shared with these applications, together with the additional inferences that could be possibly drawn using the same data leads to a variety of privacy concerns. This paper proposes SensorSafe, an architecture for managing personal sensory information in a privacy-preserving way. Our architecture consists of multiple remote data stores and a broker so users can retain the ownership of their data and management of multiple users can be well supported. SensorSafe also provides a context-aware fine-grained access control mechanism by which users can define their own sharing rules based on various conditions including context and behavioral status. We discuss our design of the SensorSafe architecture and provide application examples to show how our system can support user privacy.

**Keywords:** Information Privacy, Personal Sensory Information, Data Management Architecture

## 1 Introduction

Mobile smartphones and body-worn sensors have enabled the continuous collection of sensory information about individuals as they live their daily lives. Current smartphones are typically equipped with GPS, WiFi, and accelerometer which can provide location and activity information. Wearable sensors such as BioHarness BT [7] include ECG, respiration, and skin temperature sensors. A variety of inferences can be made by applying machine learning algorithms on the collected data. For example, stress and smoking behaviors can be detected from ECG and respiration data [31], current transportation mode can be determined by using GPS and an accelerometer [33], and personal exposure to pollutants can be measured by using location data together with a public pollutant map [28]. Collection of such sensor data and inferences have other useful applications in areas such as medical behavioral studies [31, 1], personal health-care [6, 22], and location sharing social applications [36].

An important aspect of such applications is that they involve *sharing* of personal sensitive information, which raises significant concerns about an individual's privacy. In behavioral studies, participants share their data with researchers or doctors. Location sharing applications involve friends or family members. Personal health-care applications have coaches who give useful advice about the user's health [6]. Medical home-care systems can involve not only the patient's doctor but also the insurance company [22]. Sharing of personal data is inevitable because it is essential for the application to work. While users want to share some of their personal information to benefit from certain services, they do not want to share information that they feel uncomfortable sharing.

Especially in medical behavioral studies involving multiple institutions [1], Institutional Review Board (IRB) requires that data collected from human participants should be hosted by the institution conducting the data collection [3]. Therefore, it is not possible to have a single centralized server to store data from multiple institutions. Instead, each institutional server should store its own data and interact with other institutional servers to share the data. A data management framework should be able to support such kind of IRB regulations.

A recent user study on awareness about privacy implications of sensory information [32] reports how users' privacy concerns change with personal stake and abstraction levels. In the study, users live their daily lives while wearable sensors and smartphones collect information about exercise, places, conversation, commuting, and stress. In general, the better users understand their data and the personal stakes associated with it, the higher are the concerns regarding its sharing. Especially, certain types of information such as conversation, commuting, and stress lead to more concerns than exercise and places. Privacy concerns also increase as the data contains more specific information such as place, duration, and timestamp. The fact that users have different levels of concerns depending on the types of information shared and the levels of abstractions motivates the need for a privacy-preserving data sharing framework.

Sharing sensory information poses a new challenge in protecting an individual's privacy. Traditional privacy research has focused on preventing de-anonymization of published personal data. This research has been motivated by several incidents such as de-anonymization attacks to Netflix data [29] and the AOL search records [19]. Several privacy metrics [35, 25, 24] have been proposed, and mechanisms to achieve certain privacy requirements have been devised [16]. While the traditional research protects an individual's identity, sharing of sensory information requires protection of an individual's *behavior*. That is, users want to have complete control over what kind of behavioral information is shared when they provide their sensor data. In addition, identity is often essential information in applications such as behavioral studies or health-care systems. Therefore, protecting private behaviors in sensory information becomes an important issue.

In this paper, we propose SensorSafe, an architecture that enables sharing of personal sensory information in a privacy-preserving way. Our architecture provides users with the ability to control the amount of behavioral information they want to share. This control is achieved by a context-aware, fine-grained

access control mechanism which provides numerous options to support various privacy preferences of users. Moreover, the SensorSafe architecture stores sensor data in multiple distributed servers such that users or institutions can have the ownership of their own data. Because data are not stored in a centralized server, managing data from number of users is a non-trivial problem. SensorSafe supports multiple users with ditributed data storage by having a separate broker server.

The rest of this paper is structured as follows. Related work is discussed in Section 2 and important design considerations are presented in Section 3. In Sections 4 and 5, we provide an overview and details of our architecture. We discuss application examples in Section 6 and conclude in Section 7.

## 2 Related Work

Several privacy breaches have been published [29, 19, 30], and these incidents have lead to research in protecting privacy of users when their information is shared. In an effort to protect an individual's identity in context of relational data, several privacy metrics such as k-anonymity [35], l-diversity [25], and t-closeness [24] have been proposed. Algorithms to achieve certain privacy requirements defined by the privacy metrics are also proposed. They include perturbation, suppression, generalization, and so on [16]. These research efforts mainly deal with protection against de-anonymization attacks on personal data sets. Although the data sets do not contain explicit identifiers (e.g., name, social security number), the de-anonymization attacks exploit quasi-identifiers (e.g., zip code, age, gender) which cannot be removed due to utility of the data sets. While these privacy metrics and algorithms are useful in context of relational data, they cannot be directly applied to sensory information due to several reasons. First, sensor data are often both sensitive and quasi-identifying so it is harder to anonymize without degrading much of its utility. Second, sensory information often need to be shared with identity (e.g., health-care application, medical behavioral studies) [32].

In order to protect identity when sharing sensory information, many techniques have been proposed. Several works try to preserve aggregated information by modifying original sensor data. AnonySense [14] has a *mix-network* in their architecture which anonymizes sensor data from multiple users. PoolView [17] is an architecture with perturbation scheme which adds noise to original sensor data but maintains a community average and distribution. Later, Ahmadi et al. [8] proposed a data transformation scheme which preserved a regression model of the original data. There are also several techniques for protecting identity that can be inferred from location information. Hoh et al. [20] achieve $k$-anonymous location updates using a temporal cloaking scheme. Krumm et al. [23] introduced techniques such as deleting, rounding, and addition of noise for obfuscating home location. Although these works protect privacy of identity, they do not deal with privacy of behavioral information in sensor data.

In online social networks, several architectures provide users with control over their own data. Caceres et al. [11] proposed *Virtual Individual Servers* that allows users to retain ownership of their data and to determine what data is shared with whom. PrPl [34] is a decentralized social networking infrastructure with personal data storage called *Personal Cloud Butler*. Lockr [37] provides an access control mechanism based on digitally signed social relationships. Persona [9] provides an access control mechanism via attribute-based encryption with out-of-the-band key exchange. These architectures provide access control that determines who has access to what, but more fine-grained way of access control is needed when sharing sensory information. Fine-grained access control is proposed in Locaccino [36]. However, they lack access control based on a user's context or behavior which is important to protect privacy of behavioral information in sensor data. Commercial software such as Microsoft Health Vault [5] or Google Health [2] also provide sharing of personal data with privacy in mind. However, they are designed for sharing Personal Health Records [38] rather than sensory information.

Mun et al. proposed Personal Data Vault (PDV) [27], which is an individual data storage with fine-grained access control mechanism, privacy rule recommender, and trace audit. Our system enhances the fine-grained access control by supporting privacy rules with context/behavior conditions and control for levels of inferences. In addition, while PDV is a single personal data storage, our architecture facilitates management of multiple individual data stores by having a broker server.

## 3 Design Considerations

In this section, we discuss several important considerations that have guided us to design SensorSafe. The key functionalities of SensorSafe are storing personal sensory information of *data contributors* who are willing to provide their data and sharing those data with *data consumers* who are interested in such information. The following design considerations are essential for our system to be privacy-preserving, practically usable, and effective.

**Selective sharing:** Even though data contributors are willing to share data, they do not want to share all their data because some of the collected information might disclose private and sensitive aspects of their lives. Therefore, our system provides a mechanism for data contributors to share only what they want to share. The key challenge is how a selective sharing mechanism can fully support a data contributor's various privacy preferences. SensorSafe achieves this goal by providing numerous options such as context/behavior, consumer identity, location, time, and levels of inferences.

**Data ownership:** Traditional data collection systems store the data in a centralized server [13, 28, 26]. However, the central server is the single point of failure because if the server is compromised, all the data contributors' information is at risk. Moreover, the IRB regulation mentioned in Section 1 requires multiple institutional servers. The SensorSafe architecture is distributed so the

actual storage points can be data contributors' personal computers or institutional servers.

**Managing multiple data contributors:** Each contributor's data is stored at a different physical server, making the task of data management across contributors a challenging one. Without architectural support, data consumers will have to directly contact each data contributor's server and manually manage them. Especially in behavioral studies, researchers need to search for data contributors with suitable privacy preferences because some contributors might not share enough data for the study. In SensorSafe, we have a dedicated server for managing multiple data contributors and searching for data contributors with suitable privacy rules.

**User controllable privacy:** Studies have shown that users often have difficulty expressing and managing their own privacy policies [15]. Moreover, as data collection at the contributors occurs as they live their daily lives, the privacy preferences tend to change over time. Thus, it is necessary to provide a user-friendly interface that allows the data contributors to control their information at all time. To achieve this, we have designed a web-based user interface where the users can define and manage privacy rules. The user interface consists of standard HTML UI components and Google Maps, which most Internet users are already familiar with.

**Data-store functionality:** It is important for data-storage systems to be general enough to support a variety of applications. Therefore, it should provide enough functionality to support applications. First, a data retrieval mechanism should not limit kinds of queries that applications can issue. Second, data storage should be able to store various types of data that applications require. To achieve this, SensorSafe provides expressive data query language and does not have restrictions on the structure of data.

## 4 Architecture Overview

The overall system architecture is presented in Figure 1. SensorSafe consists of remote data stores and a broker, which interact with data consumers and data contributors. Data contributors carry smartphones and possibly wearable sensors on their body. The smartphones upload collected sensor data to the data contributors' remote data stores. The data contributors create privacy rules on their data store. When the data contributors are first registered on their data store, they are automatically registered on the broker, too. Data consumers interact with the broker to search for data contributors with suitable privacy rules. After obtaining a list of the data contributors, data consumers directly communicate with remote data stores to download pertinent data. In this process, access to the data is controlled by the data contributors' privacy rules. The broker is not a performance bottleneck because sensor data are directly transferred from each remote data store to data consumers.

Figure 2 illustrates the design of remote data stores and the broker. Every interaction with both servers has to go through the user authentication layer to

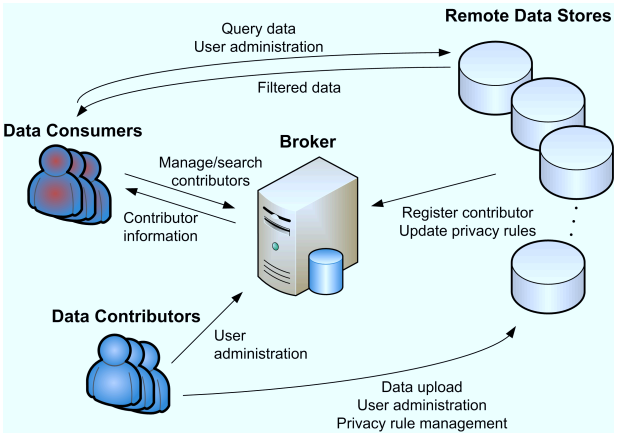
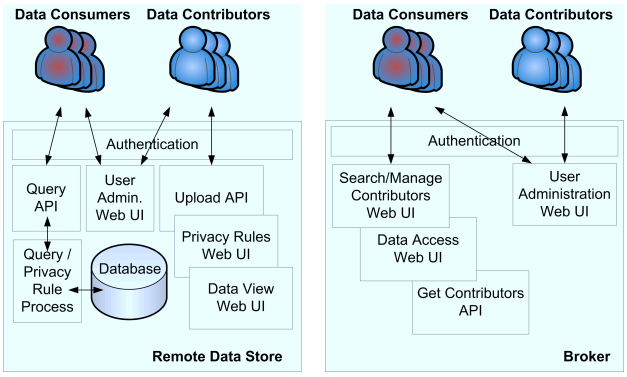



**Fig. 1.** SensorSafe Architecture




(a) Remote Data Store (b) Broker

**Fig. 2.** Remote Data Store and Broker

limit access to registered users only. Both servers also have a web user interface for user administrations. Data consumers access a contributor's data through query API provided by remote data stores. Every access is regulated by the query/privacy processing module, which interacts with the underlying database. Data contributors upload their sensor data through upload API, create/manage their privacy preferences, and view their own data using the web-based interface. Data consumers use the web interface provided by the broker to search and manage data contributors. They can also access a contributor's data through the web user interface. A list of data contributors and their data store locations can be obtained by API on the broker. The architectural details of SensorSafe are discussed in the following sections.

## 5 SensorSafe Framework

The main functionalities of SensorSafe include remote data stores, the broker, a context-aware fine-grained access control, data contributor management/searching, and privacy rule-aware data collection. Each component of SensorSafe is discussed in detail as follows.

### 5.1 Remote Data Stores

Traditional sensor data collection systems [13, 28, 26] store users' data in a centralized server. Although the centralized approach is simple and straightforward, it has several disadvantages in terms of privacy. First, data is stored in the server, which users may not trust. Second, when the centralized server is compromised, every user's data on the server is breached at the same time. Moreover, as mentioned in Section 1, the IRB regulation requires each institution stores its own data so we need multiple institutional servers. By having remote data stores, SensorSafe supports multiple institutional servers as well as personal data storage that users can trust. The storage can be a personal computer in a user's house, or the institution that collects data can provide a virtual machine pool of individual data stores and make each virtual machine accessible by its owner only. This approach allows users to store data on their own server and reduces the risk of server compromise. The advantages of virtual individual servers are also discussed in [11, 34].

**Context-aware Fine-grained Access Control** Each remote data store provides an access control mechanism. Continuously collected sensor data contain considerable amount of privacy-sensitive information about the data contributors themselves. Location and timestamp information can reveal presence in sensitive places or even life patterns. Sensors such as accelerometers, ECGs, and respiration sensors can tell a lot about contributors' activities and physiological status. Especially when sophisticated inferences are performed, more sensitive information can be revealed such as stress, smoking, conversation [31], or transportation modes [33]. Although data contributors voluntarily participate to share their data, sharing too much information increases their privacy concerns. This concern further increases when contributors understand what kinds of inferences can be drawn from their sensor data [32]. Therefore, we need a mechanism which enables contributors to share only what they want.

In order to support a variety of privacy preferences, our access control mechanism provides various conditions such as data consumer, location, time, sensor, and context. Based on the conditions, contributors can specify actions such as allow, deny, or modify the level of data abstraction. The context condition allows contributors to define privacy rules such as "don't share any data while I am driving." or "don't share data while I am in conversation." Using the conditions and actions, data contributors define a set of rules which express their privacy preferences and the remote data store enforces the rules. Table 1 summarizes the conditions and actions of privacy rules.

**Table 1.** Various Options for Privacy Rules

(a) Conditions and Actions

| Options | | Attributes |
|---|---|---|
| Conditions | Data Consumer | User Name, Group Name, Study Name |
| | Location | Pre-defined Label, Region Coordinates |
| | Time | Time Range, Repeated Time |
| | Sensor | Sensor Channel Name (e.g., Accelerometer, ECG) |
| | Context | Available context from sensors (e.g., Moving, Not Moving, Still, Walk, Run, Bike, Drive, Stress, Conversation, Smoke) |
| Actions | | Allow, Deny, Abstraction |

(b) Example Abstraction Options

| Context | Options |
|---|---|
| Location | Coordinates, Street Address, Zipcode, City, State, Country, Not Share |
| Time | Milliseconds, Hour, Day, Month, Year, Not Share |
| Activity | Accelerometer Data, Still/Walk/Run/Bike/Drive, Move/Not Move, Not Share |
| Stress | ECG/Respiration Data, Stressed/Not Stressed, Not Share |
| Smoking | Respiration Data, Smoking/Not Smoking, Not Share |
| Conversation | Microphone/Respiration Data, Conversation/Not Conversation, Not Share |

*Basic conditions:* Using the data consumer condition, contributors can specify who will be affected by this privacy rule. It can be a unique user name of a data consumer, or a group or study name which includes a set of data consumers. Data contributors specify locations by defining a region on a map user interface. Time condition is defined as a continuous time range (e.g., from Feb. 2011 to Mar. 2011) or repeated time (e.g., 3-6pm on every Wednesday). Using the sensor condition, contributors can select specific sensor channels in their privacy rules.

*Context condition:* Data contributors can also define their privacy rules using context information drawn from sensor data. For example, microphones and respiration sensors can be used to infer whether a data contributor is in conversation or not. An accelerometer with GPS can provide transportation information such as walking, running, biking, and driving. A data contributor might not feel

comfortable sharing sensor data while in conversation with someone or while driving. In these cases, context conditions can be used to describe such privacy rules.

*Actions:* Data contributors can either allow or deny access to sensor data which satisfy the basic and context conditions. When allowed, raw sensor data are shared with corresponding data consumers. In addition, contributors can share more abstracted information instead of sharing raw sensor data. For example, instead of sharing latitude and longitude coordinates, contributors can abstract this information as zip code, city, or state names. With accelerometers, contributors can choose to share only transportation modes (e.g., still, walk, run, bike, drive) or just whether moving or not moving. Note that a sensor can be used to infer multiple context information (e.g., a respiration sensor is used for stress, conversation, and smoking). Therefore, if a contributor chooses not to share such a sensor or a related context, the raw sensor data will not be shared even though other relevant contexts are chosen to be shared in raw data form. For example, if the smoking context is not shared, respiration sensor data will not be shared even though stress and conversation are shared in raw data form. This is because once respiration data are provided by stress or conversation, smoking can be also inferred from the data. The privacy rule processing module contains this sensor/context dependency information and performs access control accordingly.

Privacy rules are created and edited using a web user interface shown in Figure 3. The user interface consists of Google Maps, calendars, dialog boxes, and common HTML UI components such as text boxes, check boxes and radio buttons. Although a usability study of our user interface remains as future work, we believe users will find it easy to create privacy rules because most Internet users are familiar with our web user interface components. Once data contributors define their privacy rules using the web UI, they are stored as JSON objects [4] on the remote data stores. Figure 4 shows an example privacy rule with its corresponding JSON representation.

**Data Storage** *Wave Segments:* Another important aspect of a remote data store is that it needs to handle large volumes of data generated by continuous sensing. Storing the time series of sensor data as individual tuples is inefficient both in terms of storage size and querying time. Therefore, in order to increase scalability and computational efficiency, it is essential for a remote data store to have a compact representation of data.

In a remote data store, a continuous stream of sensor data is divided into many segments, called *wave segments*, an extension of an abstract data type proposed in [18]. A wave segment is the smallest unit of data representation, and each wave segment typically contains hundreds or thousands of data samples. We further observe that sensors are typically sampled in uniform intervals, so a wave segment stores timestamp information as a start time and a sampling interval. A wave segment can also have an individual timestamp for each data

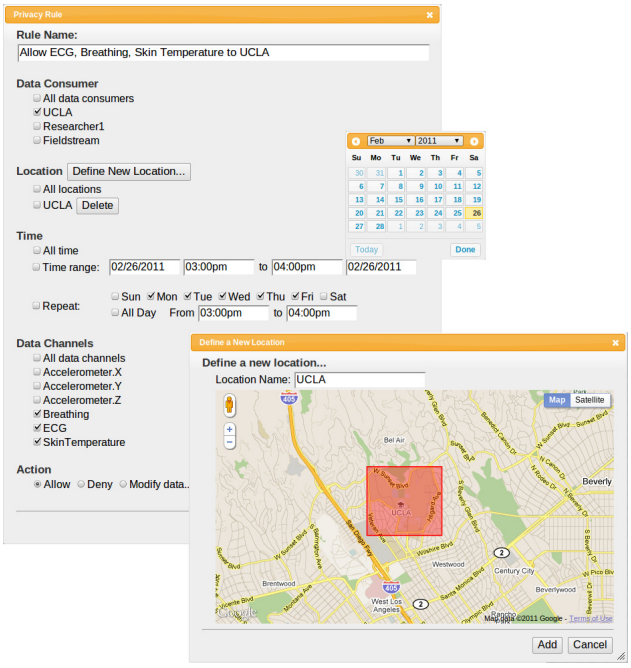



**Fig. 3.** Web User Interface for Privacy Rules

sample, which is necessary to represent sampling schemes such as adaptive [21], compressive [12], and episodic.

Sequences of data samples from multiple sensor channels are typically stored as Binary Large Objects (blob) in database systems. A wave segment also consists of a sensor value blob and additional metadata describing the value blob. The value blob is an array of tuples each containing values from multiple sensor channels. The metadata includes a start time, a sampling interval, a location, and a format of tuples in the value blob. For non-periodic sampling and mobile sensors, time and location stamps are stored in the value blob as additional sensor channels. Figure 5 shows an example of a wave segment represented in Java Script Object Notation (JSON) [4].

*Wave Segment Optimization:* The number of wave segments directly affects query performance because it is the number of records stored in a database. Therefore, it is important for each wave segment to include a large enough number of samples. Because memory space is constrained at the sensors, a single data packet from the sensors typically contains dozens or hundreds of samples. For example, the Zephyr chest-band transmits 64 ECG samples in a single packet [7]. If this packet is directly converted to a wave segment, there will be too many wave segments in total decreasing the query performance. Therefore, remote data stores perform a wave segment optimization by merging them as much as pos-

```
[{  'Consumer': [ 'Bob' ],
   'LocationLabel': [ 'UCLA' ],
   'Action': 'Allow'
},
{  'Consumer': [ 'Bob' ],
   'LocationLabel': [ 'UCLA' ],
   'RepeatTime': { 'Day': [ 'Mon', 'Tue', 'Wed', 'Thu', 'Fri' ],
                   'HourMin': [ '9:00am', '6:00pm' ] },
   'Context': [ 'Conversation' ]
   'Action': { 'Abstraction': { 'Stress': 'NotShared' } }
}]
```

**Fig. 4.** Example of JSON Privacy Rule: "Share all data collected at UCLA with Bob but do not share stress information while I am in conversation at UCLA on Weekdays from 9am to 6pm."

```
{ 'StartTime': null,
  'SamplingInterval': null,
  'StaticLocation': null,
  'ValueBlobFormat': [ 'Time', 'X', 'Y', 'Z',
                       'AccX', 'AccY', 'AccZ' ],
  'ValueBlob': [
     [ 1267662001.752, -118.44304025173187,
       34.069381356239319, null, 1899, 1993, 1614 ],
     [ 1267662001.754, -118.44304025173187,
       34.069381356239319, null, 1900, 1985, 1617 ],
     [ 1267662001.756, -118.44304025173187,
       34.069381356239319, null, 1898, 1990, 1621 ],
     ...
  ]
}
```

**Fig. 5.** Example of a Wave Segment in Java Script Object Notation

sible. If timestamps of two wave segments are consecutive, they can be merged as long as they have the same location coordinates and data channels.

### 5.2 Broker

In scientific behavioral studies, data consumers (study coordinators such as researchers or doctors) typically recruit many data contributors (study participants) [1, 31]. In participatory sensing [10], a data collection campaign also involves many data contributors. In centralized systems, it is trivial to manage those multiple data contributors because every data contributor is registered on a single server. However, if storage for the data contributors are distributed and there is no dedicated server to maintain the list of the data contributors, it is not trivial to manage all the individual data stores.

**Data Contributor Management** Therefore, SensorSafe has a broker that manages all the remote data stores so the data consumers can easily access them. The broker stores every data contributor's identity and the IP address of

the associated remote data store. Using the web user interface on the broker, the data consumers can create a list of data contributors or search for suitable data contributors. Data contributor searching is further discussed in the following section. The broker also provides a convenient web user interface for accessing contributors' data. The web interface provides query options such as location, time, and data channels so the data consumers can retrieve data in which they are interested. Data consumer applications also can obtain a list of data contributors and their IP addresses by using an API provided by the broker.

**Data Contributor Searching** From the data consumer's point of view, a data contributor's privacy rules directly affect the utility of the sensor data. Depending on the privacy rules, a contributor's data could be partially useful for data consumers or not useful at all. For example, suppose a data consumer is interested in studying stress events and related physiological signals in work environments. However, a data contributor participating in the study defines a privacy rule saying he/she does not want to share stress-related data at work place. In this case, these data are not useful to the data consumer. The data consumer needs to find other contributors who share enough data for the data consumer's study.

In SensorSafe, the broker provides a web user interface for searching for data contributors with suitable privacy rules so that data consumers can find contributors who share enough data for their study. The broker locally stores all privacy rules of every user on remote data stores to search through them. Whenever data contributors change their privacy rules, remote data stores automatically communicate with the broker to synchronize the privacy rules. Data consumers can search for all conditions and actions of privacy rules such as location, time, sensor, context, and abstraction. For example, finding data contributors who share ECG and respiration sensor data at the location labeled "work" from 9am to 6pm on weekdays can be performed. After searching suitable data contributors, data consumers can store the list of contributors to access their data.

### 5.3 Privacy Rule-Aware Data Collection

If a privacy rule says not to share data at a certain location, time, or context, it is better not to collect such data in the first place because the data will not be shared anyway. In order to enable this, smartphones carried by data contributors download the owner's privacy rules from the remote data stores and determine whether to collect data based on the privacy rules. The decision can be made on three conditions such as current location, time, and context. When there are no data to be shared at the current location and time, sensors will be disabled. In case of a context condition, sensor data are first temporarily collected on a smartphone to infer current context. If there are no data to be shared in the current context, the data will be discarded.

Although privacy rule-aware data collection provides a more secure way to collect data, but one should not overlook cases in which a data contributor wants

to change privacy rules after collecting data. If a contributor wants to share data that have not been collected at all, there is no way to recover them. Therefore, we provide privacy rule-aware data collection as optional functionality, and data contributors need to carefully decide whether or not to use this option.

### 5.4 Authentication

When data consumers and contributors access the broker and remote data stores through APIs, they are authenticated by their unique API keys. An API key is a random string generated by the SHA algorithm. Each user obtains a unique API key when he/she is first registered to the servers. Users must keep their API keys private because it acts as a username and a password. In order to secure the API key during communications, it is included in the body of a HTTPS POST request and the communication is secured with HTTPS. When a data consumer first accesses a certain data contributor's remote data store, he/she needs to register to the remote data store and obtain an API key. Therefore, a data consumer might have many API keys for multiple remote data stores. However, the registration process is automatically handled by the broker and the list of API keys are stored on the broker. Data consumer applications use the HTTP API on the broker to obtain a list of data contributors with corresponding remote data stores and API keys. Accesses to web user interfaces are authenticated by a login system using a username and a password.

## 6 Application Examples

In this section, we show how SensorSafe can support sharing sensor data in privacy-preserving way. Our example scenarios include two applications: a medical behavioral study and a health-care application. In the behavioral study, researchers want to analyze effects of various environmental factors on an individual's stress level [31]. For this study, data contributors wear a chest band equipped with an ECG and a respiration sensor. They also carry smartphones which record acceleration, time, location, and voice on the microphone. They live their normal life as the sensor data are collected automatically. On the smartphone, various contextual information such as stress, smoking, conversation, and transportation modes are inferred using the sensors on the phone and the chest band. The sensor data are annotated with the context information and uploaded to remote data stores. In our health-care application scenario, data contributors want to share their daily activities with personal coaches to get advice on exercise and health habit [6].

A data contributor, say Alice, first decides to share all data with the researchers. After logging into her remote data store, she defines a privacy rule that allows the researchers to access all the data. She also thinks her health coach only needs activity data so she defines a privacy rule that allows the health coach to access accelerometer data only. After collecting data for one day, Alice reviews her data using the web user interface on her data store. Alice finds out that she

is frequently stressed while driving. She feels uncomfortable with sharing this information so she adds a privacy rule that denies access to stress data while driving. She also feels uncomfortable sharing activity data while she is home so she adds a privacy rule which denies accelerometer data collected at her home location. She is certain that she is not going to change her mind about sharing her data so she turns on privacy rule-aware data collection on her smartphone. Whenever the smartphone detects she is driving, it stops collecting ECG and respiration data which are related to stress inference. Whenever the smartphone detects that current location is her home, it also stops collecting accelerometer data.

A data consumer, say Bob, wants to obtain some data for his study. He has recruited 20 data contributors including Alice. He first logs into the broker server and adds the data contributors to his account. When he adds his data contributors, the broker automatically registers Bob to the remote data stores to obtain an API key. Bob is especially interested in people's stress behavior while they are driving. Because Bob knows that not every data contributor will share their stress information while driving, he uses a data contributor searching function on the broker. After searching for suitable data contributors, he obtains a list of data contributors without Alice and saves the list in his account. Bob reviews the contributors' data using the web user interface provided on the broker. Bob also uses his own data analyzing software. The software first obtains the list of data contributors with access information for their remote data stores from the broker. Then, the software downloads the contributors' data using the query API provided by each remote data store.

## 7 Conclusion and Future Work

This paper presents SensorSafe, a framework that enables data contributors to share their personal sensory information with data consumers in privacy-preserving way. Using its context-aware fine-grained access control mechanism, data contributors can define sharing rules with various conditions including current context or behavior. To improve the usability of SensorSafe, we implemented web-based user interfaces for defining privacy rules and reviewing a user's data. In addition, data contributors retain the ownership of their data by using remote data stores. SensorSafe also supports applications involving data collection from multiple contributors and institutions by having a separate broker server. We provide application examples to show how SensorSafe can support user privacy and management of sensory information. In the future, user studies will be conducted to evaluate and improve the usability of our system and also provide understanding of how people share personal data with others. Moreover, in order to improve security of the SensorSafe architecture, we will analyze our system for various attack scenarios and implement appropriate security mechanisms.

## References


1. FieldStream: network data services for exposure biology studies in natural environments. http://www.fieldstream.org/.
2. Google health. http://www.google.com/intl/en/health/about/.
3. Institutional Review Board - Protect Research Data. http://irb.ufl.edu/irb01/data.html.
4. JavaScript object notation. http://www.json.org/.
5. Microsoft HealthVault. http://www.healthvault.com.
6. Philips DirectLife: fitness, health and successful weight management. http://www.directlife.philips.com/.
7. Zephyr technology corporation, BioHarness BT. http://www.zephyr-technology.com/bioharness-bt.
8. H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, and J. Han. Privacy-aware regression modeling of participatory sensing data. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 99–112, 2010.
9. R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. *ACM SIGCOMM Computer Communication Review*, 39(4):135–146, 2009.
10. J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava. Participatory sensing. In *World Sensor Web Workshop*, pages 1–5, 2006.
11. R. Cáceres, L. Cox, H. Lim, A. Shakimov, and A. Varshavsky. Virtual individual servers as privacy-preserving proxies for mobile devices. In *In Proc. of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, pages 37–42, 2009.
12. E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.
13. K. Chang, N. Yau, M. Hansen, and D. Estrin. Sensorbase.org: a centralized repository to slog sensor network data. In *Proc. of the International Conf. on Distributed Networks(DCOSS)/EAWMS*, 2006.
14. C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. AnonySense: Privacy-aware people-centric sensing. In *Proc. of the 6th international conference on Mobile systems, applications, and services*, pages 211–224, 2008.
15. J. Cornwell, I. Fette, G. Hsieh, M. Prabaker, J. Rao, K. Tang, K. Vaniea, L. Bauer, L. Cranor, J. Hong, et al. User-controllable security and privacy for pervasive computing. In *Proc. of the 8th IEEE Workshop on Mobile Computing Systems and Applications*, pages 14–19, 2007.
16. B. C. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey on recent developments. *ACM Computing Surveys*, 2010.
17. R. Ganti, N. Pham, Y. Tsai, and T. Abdelzaher. PoolView: stream privacy for grassroots participatory sensing. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 281–294, 2008.
18. L. Girod, Y. Mei, R. Newton, S. Rost, A. Thiagarajan, H. Balakrishnan, and S. Madden. XStream: a Signal-Oriented Data Stream Management System. In *Proc. of IEEE 24th International Conference on Data Engineering*, pages 1180–1189, 2008.
19. S. Hansell. AOL removes search data on vast group of web users. New York Times, Aug 8, 2006.

20. B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *Proc. of the 6th International Conference on Mobile Systems, Applications, and Services*, pages 17–20, 2008.
21. A. Jain and E. Chang. Adaptive sampling for sensor networks. In *Proc. of the 1st international workshop on Data management for sensor networks: in conjunction with VLDB 2004*, pages 10–16, 2004.
22. D. Kotz, S. Avancha, and A. Baxi. A privacy framework for mobile health and home-care systems. In *Proc. of the first ACM Workshop on Security and Privacy in Medical and Home-care Systems*, pages 1–12, 2009.
23. J. Krumm. Inference attacks on location tracks. *Pervasive Computing*, pages 127–143, 2007.
24. N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: privacy beyond k-Anonymity and l-Diversity. In *Proc. of IEEE 23rd International Conference on Data Engineering*, pages 106–115, 2007.
25. A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, Mar. 2007.
26. E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proc. of the 6th ACM conference on Embedded network sensor systems*, pages 337–350, 2008.
27. M. Mun, S. Hao, N. Mishra, K. Shilton, J. Burke, D. Estrin, M. Hansen, and R. Govindan. Personal data vaults: a locus of control for personal data streams. In *Proc. of the 6th International Conference*, page 17, 2010.
28. M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proc. of 7th International Conference on Mobile Systems, Applications, and Services*, pages 55–68, 2009.
29. A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Proc. of IEEE Symposium on Security and Privacy*, pages 111–125, 2008.
30. A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proc. of IEEE Symposium on Security and Privacy*, pages 173–187, 2009.
31. K. Plarre, A. Raij, S. M. Hossain, A. A. Ali, M. Nakajima, M. al'Absi, E. Ertin, T. Kamarck, S. Kumar, M. Scott, D. Siewiorek, A. Smailagic, and L. E. Wittmers Jr. Continuous inference of psychological stress from sensory measurements collected in the natural environment. In *Proc. of 10th International Conference on Information Processing in Sensor Networks*, 2011.
32. A. Raij, A. Ghosh, S. Kumar, and M. Srivastava. Privacy risks emerging from the adoption of innocuouswearable sensors in the mobile environment. In *Proc. of ACM CHI Conference on Human Factors in Computing Systems*, 2011.
33. S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Determining transportation mode on mobile phones. In *Proc. of 12th IEEE International Symposium on Wearable Computers*, pages 25–28, 2008.
34. S. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. Teh, R. Chu, B. Dodson, and M. Lam. PrPl: a decentralized social networking infrastructure. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, pages 1–8, 2010.
35. L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10:557–570, Oct. 2002.

36. E. Toch, J. Cranshaw, P. Hankes-Drielsma, J. Springfield, P. G. Kelley, L. Cranor, J. Hong, and N. Sadeh. Locaccino: a privacy-centric location sharing application. In *Proc. of 12th ACM International Conference on Ubiquitous Computing*, pages 381–382, 2010.
37. A. Tootoonchian, S. Saroiu, Y. Ganjali, and A. Wolman. Lockr: better privacy for social networks. In *Proc. of the 5th international conference on Emerging networking experiments and technologies*, pages 169–180, 2009.
38. Wikipedia. Personal health record. http://en.wikipedia.org/wiki/Personal_health_record.