Pervasive and Mobile Computing 8 (2012) 331–345

Contents lists available at SciVerse ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Fast track article

Balancing behavioral privacy and information utility in sensory data flows

Supriyo Chakraborty*, Zainul Charbiwala, Haksoo Choi, Kasturi Rangan Raghavan, Mani B. Srivastava

University of California, Los Angeles, United States

ARTICLE INFO

Article history:
Available online 20 March 2012

Keywords:
Behavioral privacy
Trust
Information sharing
Sensory data

ABSTRACT

Miniaturized smart sensors are increasingly being used to collect personal data which embed minute details of our everyday life. When shared, the data streams can easily be mined to draw a rich set of inferences regarding private behaviors and lifestyle patterns. Disclosure of some of these unintended inferences gives rise to the notion of *behavioral privacy* different from traditional *identity privacy* typically addressed in the literature. From the provider's perspective, we summarize these privacy concerns into three basic questions: (i) Whom to share data with? (ii) How much data to share? and (iii) What data to share?

In this paper, we outline the architecture of SensorSafe as a software-based framework with support for three basic mechanisms to allow privacy-aware data sharing. First, it provides a library of routines accessible using a simple GUI for providers to define fine-grained, context-dependent access control. Second, it uses the trust network between consumers and providers to derive the optimal rate of information flow which would maintain both provider privacy and consumer utility. Finally, it introduces a compressive sensing based feature-sharing procedure to further control the amount of information release. We provide simulation results to illustrate the efficacy of each of these mechanisms.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

As personal spaces are being increasingly instrumented through smart and miniaturized sensors, unobtrusive acquisition of personal data has made the information privacy problem more relevant than ever before. The canonical privacy problem as stated for population scale databases can be summed up as: given a collection of personal records from individuals, how would one disclose either the data or "useful" function values such as correlations or aggregate population characteristics computed over the data, without revealing *any* individual information. This notion of absolute privacy is analogous to the principle of semantic security formulated for crypto-systems in [1]. A hypothetical "privacy-first" approach to the above problem would be to disclose no data or random data bearing no relation to the actual database. However, the explicit *utility* requirements from the disclosed information prohibits the use of such a scheme. In addition, the utility requirement in conjunction with adversarial access to externally available auxiliary information makes it impossible to achieve absolute privacy [2]. Current research in database privacy has thus evolved into a study of the trade-offs involving degradation in the quality of information shared, owing to privacy concerns, and the corresponding effect on its utility [3,4].

* Corresponding author.
E-mail addresses: supriyo@ucla.edu, supriyo.chakraborty@gmail.com (S. Chakraborty), zainul@ucla.edu (Z. Charbiwala), haksoo@ucla.edu (H. Choi), kasturir@ucla.edu (K.R. Raghavan), mbs@ucla.edu (M.B. Srivastava).

1574-1192/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2012.03.002

While privacy of identity [5–7], or of specific data attributes [2,4] has been extensively studied, the proliferation of smartphones with embedded and wireless connected wearable sensors presents a different kind of privacy problem which we term as *behavioral privacy*. The miniaturized sensors, ported by individuals as they perform their daily activities, collect large amount of personal data. This data is more often than not shared with a variety of applications such as for crowd-sourcing [8,9], environment monitoring [10], healthcare and behavioral studies [11–13], carbon exposure tracking [14], traffic estimation [15,16], and smart grid monitoring [17]. These applications in turn provide services in the form of both population- and individual-level inferences. However, the sensory data shared is really our digital footprint, embedding in it minute details of our everyday lives. Mining it in conjunction with externally available auxiliary information allows an adversary to draw a rich set of unintended private behavioral inferences. For example, physiological data shared for healthcare studies can also be used to infer addictions like smoking [13], location traces for traffic estimation reveal travel routes, frequently visited places [18], smart meter data can be used to reveal personal habits [19], occupancy [20] and so on.

Increased awareness of the stakes involved in sharing personal data, therefore, gives rise to behavioral privacy concerns that hinder user participation. Users choose to deliberately transform or obfuscate data to preserve privacy but when done arbitrarily, this obfuscation leads to a reduction in application utility, or worse, database poisoning. From a data provider's (user's) perspective the fundamental privacy questions while sharing sensory data are: (1) What data should be shared and with whom? (2) How much data should be shared? and (3) Can the data be transformed such that privacy requirements and data consumer's (application's) utility needs are simultaneously satisfied? The privacy requirement of the provider and the utility objectives of the consumer creates a tension which is at the crux of the behavioral privacy problem.

1.1. Privacy: reality or myth

Is privacy something that people really care about? Recent surveys in [18,21] summarized interesting opinions about privacy from multiple independent studies. While people in general were oblivious to privacy violations and amenable to sharing their data, the perception quickly morphed into one of concern when apprised of the various sensitive inferences that could be drawn and the resulting consequences.

Some of the recent high-visibility fiascos have further established privacy as a important sharing constraint. Examples include the de-anonymization of the publicly released AOL search logs [22] and the movie-rating records of Netflix subscribers [23]. The large datasets in question were released to enable data-mining and collaborative filtering research. However, when combined with auxiliary information the anonymized datasets were shown to reveal identity information of individual users.

Sharing sensory data also presents unique challenges. For example, households in the US are being equipped with smart meters to act as providers of temporally fine-grained energy consumption reports. The utility companies act as consumers of the data, and use the reports to better estimate the domestic power consumption leading to optimized distribution and control of the power grid. However, as shown in [17] and more recently reported in [19], several unintended and sensitive inferences such as occupancy and lifestyle patterns of the occupants can be made from the data in addition to total power consumption. In fact, privacy has been identified as a major challenge in fine-grained monitoring of residential spaces [20]. Similarly, in medical research the continuous physiological data collected by wearable sensors can be used to infer potentially sensitive information such as smoking or drinking habits [13], food preferences and so on. While "informed consent" of the provider is the currently used sharing policy, it can be easily overlooked causing privacy violations as exemplified in [24]. Similarly, participatory sensing applications require users to voluntarily upload their personal data for purposes of a study. For example, PEIR [14], a popular tool for computing carbon exposure levels during trips, requires users to upload their travel paths. In the absence of adequate privacy transformation, the uploaded traces could be used to find out frequently traveled paths, workplace, home and other sensitive locations.

Thus, privacy threat during data disclosure is real and unless adequate mitigation steps are taken it could cause a delay in the adoption of various ubiquitous sensing based applications.

1.2. Privacy problem characterization

The various privacy threats could be grouped into two broad classes [25]:

Identity violation: This occurs when the identity of a provider is revealed from the shared data. Typically, during database disclosures, the goal is to provide *privacy in numbers* by anonymizing released data to make the provider identity indistinguishable within a sub-population. However, privacy violation occurs when the shared attributes in association with auxiliary information are used to reveal identity (e.g. include Netflix [23] and AOL [22] fiascos). A detailed analysis of the challenges and pitfalls of data anonymization can be found in [26].

Inference violation: This occurs when the data from a provider is used to draw *unintended* inferences in addition to the *intended* ones agreed upon by the consumer during data sharing. The list of unintended inferences defines the privacy requirements of the provider. This class of violations manifests itself when the user shares personal data (e.g. data collected using body-worn sensors and smartphones) for personalized services such as remote health monitoring, stress-level monitoring, activity tracking, pollution exposure and so on. For reasons of personalization, the provider is in most of the above cases willing to reveal his identity and instead protect against a specific set of inferences that could be drawn using the shared data. For example, in a medical study ECG data is needed to monitor variability in heart rate. However, the data should not be used for other sensitive inferences such as smoking or drinking habits of the individual [13]. The attacks using smart meter data [17], location traces in PEIR [14], or the unauthorized use of DNA samples in [24] indicated in Section 1.1, fall under this category.

1.3. Our contribution

The behavioral privacy problem while sharing sensory data is an instance of the inference violation class where the provider wants to protect against specific behavioral inferences. However, as discussed earlier there are multiple aspects to the problem (who?, how much?, what?). Therefore, any solution framework needs to piece together the individual solution elements and allow the provider a unified interface for privacy-aware data sharing. In this paper, we describe SensorSafe as a framework which uses three distinct mechanisms for addressing the provider's privacy concerns. Specifically, our contributions are as summarized below:

(i) *Whom to share with?*: A provider should be able to control both the data being shared as well as the consumers who can access the data. To this end, we describe the first mechanism that provides fine-grained access control primitives and abstraction functions for data obfuscation before sharing. A provider could use these primitives to define constraints on one or multiple filters such as data attributes, context (location, sensor type) and consumer identity.

(ii) *How much to share?*: The resolution of the data shared depends on multiple factors such as the information quality requested by the consumer, the *trust* that the provider has on the consumer, as well as the trust network of the consumer. While variability in mutual trust is a precursor to privacy concerns, a consumer's trust network is indicative of the possible information leakage due to collusion. Our second mechanism uses the trust network between the producer and consumers to quantify the information leakage for a given data resolution. It then uses a linear program to find the optimal data resolution which maximizes the consumer utility and minimizes producer's information leakage.

(iii) *What to share?*: Once we have decided whom to share with and how much to share, the final step is to choose what data to share and how to obfuscate it such that no more than the desired amount of information is shared. This amount is such that the revealed inferences could be computed accurately whereas the concealed ones could not be computed. To this end, we start by formalizing the notion of behavioral inference privacy. We note that existing privacy approaches primarily designed to protect against complete data reconstruction and identity privacy no longer work for the inference violations which can occur over partially reconstructed data and when the provider identity is already revealed. Finally, we propose a compressive sensing [27] based privacy mechanism for solving the inference problem and provide preliminary results obtained using our approach.

1.4. Organization of the document

In Section 2, we present a classification of the various approaches that have been proposed for preserving privacy. In Section 3, we summarize the design and architecture of SensorSafe. In Section 4, we discuss our trust network aware data sharing scheme. This is followed by Section 5 where we discuss the formalization of the behavioral privacy problem and propose a compressive sensing based privacy mechanism. We conclude in Section 6.

2. Related work

In this section, we classify the various approaches that have been proposed for preserving privacy. For each class, we show that the techniques are either inadequate for handling the behavioral privacy problem or in their current form are infeasible for practical implementation.

2.1. Obfuscation strategies

Obfuscation can be described as an act of deliberate data transformation, performed prior to information release, for reasons of privacy preservation. Information release can occur either in a *non-interactive* or an *interactive* setting. In a non-interactive setting, the (database) provider performs a sanitization of the data by removal of personally identifiable information (PII) followed by obfuscation of the quasi-identifiable (QI) attributes to protect linkages to the sensitive attributes from the released dataset. A popular metric for data obfuscation is k-anonymity [5]. It requires that every record in the released dataset is indistinguishable from at least $k-1$ other records on the released attributes. Operations such as generalization, suppression, permutation and perturbation are performed on the attributes for achieving k-anonymity before their release. A detailed survey of these techniques could be found in [28,18]. Not only are some of these operations ad-hoc in nature, but in the absence of diversity among the selected k users, and in presence of auxiliary information, [26,29,23] have shown that de-anonymization of data is possible leading to *identity violation*. Newer metrics such as l-diversity [6] and t-closeness [7] have also been proposed to thwart de-anonymization attacks. Roughly speaking, l-diversity requires that any group of obfuscated QI attributes has at least l distinct values for the sensitive attributes. The additional diversity does provide better protection compared to k-anonymity but still suffers from vulnerability to probabilistic inference attacks. t-closeness requires that the distance between the distributions of the sensitive attributes in the released dataset and in the entire database should be less than a pre-defined parameter t. It does not protect against identity disclosure.

In an interactive setting, the provider discloses information as responses to consumer queries. In this setting the provider has greater control over the released data. For example, in [30] providers independently perturb their data using an application-specific noise model. The noise model is such that it preserves privacy against typical *reconstruction attacks*, while allowing the consumer to compute community aggregates from the collected data. In [31], a data transformation scheme is proposed for the provider which preserves a regression model of original data. Similarly, [32] uses the concept of data slicing and mixing for preserving privacy, and allows computation of statistical additive and non-additive aggregation functions at the consumer.

Metrics such as k-anonymity and l-diversity protect against identity privacy. However, for personalization of requested service identity is typically revealed while sharing sensory data. t-closeness protects against specific data values, but does not protect against specific inferences. In addition, while [30–32] protect against full reconstruction attacks, private inferences could be made even on partially reconstructed data.

2.2. *Differential privacy*

Proposed in [2] with strong provable privacy guarantees, differential privacy aims to replace the mostly ad-hoc obfuscation strategies with a principled data release mechanism for statistical databases. To achieve differential privacy it is required that the response to a query including or excluding a particular database entry is indistinguishable in the probabilistic sense. This in turn guarantees that an adversary gains negligible information on individual records upon observing the output of a computation regardless of the auxiliary data available to him.

Formally, consider any two databases $D_1, D_2 \in \mathbb{R}^n$ that differ in exactly one entry. Let $\kappa_f(D_1)$ and $\kappa_f(D_2)$ be the responses from D_1 and D_2, respectively, where κ_f is the mechanism used to respond to an arbitrary query $f()$. A randomized mechanism κ_f provides ϵ-differential privacy if

$$\frac{P(\kappa_f(D_1) \in S)}{P(\kappa_f(D_2) \in S)} \leq e^{\epsilon} \tag{1}$$

where $S \subseteq \text{Range}(\kappa_f)$. The ratio in Eq. (1), represents the "knowledge gain" for an intruder moving from one version of the database to the other. In order to achieve differential privacy, [2,33] suggest the use of Laplace based noise addition. However, the calibration of noise magnitude to simultaneously maintain data utility while preserving privacy is an important issue that needs to be addressed for making it practically relevant [34].

Predicated on aggregate query/response systems, differential privacy protects against a specific inference which is the *membership disclosure* of an entry in a population-scale database. However, behavioral privacy is due to a set of private inferences drawn over individual (not population-scale) data streams shared by providers.

2.3. *System-based approaches*

Various system architectures, primarily for online social networks, have been proposed to help providers retain ownership and exercise greater control while sharing data. We summarize some of them here and outline how SensorSafe is built on top of these existing design principles. However, it is worth emphasizing that the access control problem addressed by these existing systems is orthogonal to the inference violation problem which arises when access control has already been applied and the consumer is in possession of the shared data.

In [35], provider-owned *Virtual Individual Servers* are proposed as proxies for uploading user generated content instead of third-party services. A decentralized social networking infrastructure with personal data storages called *Personal Cloud Butler* has been advocated in [36]. Several implementations of access control lists (ACLs) have also emerged. For example, Lockr [37] implements ACL based on social attestation and Persona [38] uses attribute-based encryption with an out-of-band key exchange. While the above systems are for social networks, Locaccino [39] supports fine-grained ACLs, needed by sensory data. 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—a feature we incorporate in SensorSafe. Finally, Personal Data Vault (PDV) [40] is a recent work which allows individual data storage, fine-grained ACLs for sensory data, implements a privacy rule recommender, along with trace audit functions tracking how the data is used by the consumer. While SensorSafe closely follows the PDV model, it enhances its fine-grained access control by having context-aware rule processing and notification for conflicting rules. PDV also provides a single personal data storage whereas SensorSafe facilitates multiple individual data stores with the broker service. In addition, it takes into account the trust network between the providers and consumers for guiding the privacy policies.

2.4. *Cryptographic techniques*

In cryptography, a pre-defined secret in the form of a decryption key is used to differentiate a valid consumer from an adversary. However, in the privacy setting, consumer and adversary are one and the same and there does not exist a secret key to set them apart. This prevents existing cryptographic solutions from being applicable to the privacy setting [2]. However, some of the recent techniques summarized below provide interesting possibilities toward adoption of a cryptographic solution for the privacy problem.

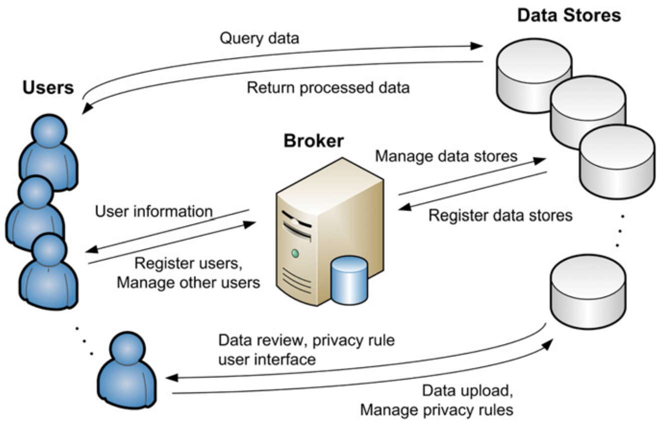

Data Stores
Query data
Return processed data
Users
Broker
Manage data stores
Register data stores
User information
Register users, Manage other users
Data review, privacy rule user interface
Data upload, Manage privacy rules

Fig. 1. SensorSafe framework.

Functional encryption: A recent technique proposed in [41] supports restricted secret keys that enable a key holder to learn a specific function of the encrypted data and nothing more about the data. Functional Encryption takes a different approach toward public key encryption. Traditionally, encryption is targeted toward a specific consumer bearing a secret key and the access to the encrypted data is all or nothing—either one can decrypt and read the entire plain text or nothing at all. Instead in functional encryption, the provider does not encrypt for a specific consumer, but only specifies how to share the data. Also a decryption key allows a consumer to learn only a function of the encrypted data. However, this technique is still in its infancy and currently supports a small number of functions (inner products only).

Homomorphic encryption: Often referred to as the holy grail of encryption, the homomorphic scheme allows one to perform computations on the cipher text itself without the need to decrypt it. Therefore the consumer never has access to the plain text effectively restricting the subset of functions (or unintended inferences) which could be computed using the shared data. A fully homomorphic scheme proposed in [42] supports the computation of any function over the encrypted data. However, the computation and storage overhead of implementing fully homomorphic encryption are limitations that need to be overcome before it could be used.

2.5. Inference sharing

Is it possible to share the inferences, or the data mining results instead of the raw data samples? In most of the cases, the models used for computing the results are proprietary to the consumer prohibiting their sharing. At other times, the provider might not be willing to incur the storage and the computational resources required for computation of the inference. In addition, the consumer might not want to reveal the inferences he would like to draw using the collected data. Finally, inference drawn using aggregate data, is more accurate when raw data is used rather than soft fusion of individual decisions.

3. Whom to share with: the sensorsafe framework

The personal sensory information collected is typically shared with various types of consumers such as doctors, medical researchers, third-party applications, cloud-based services and so on. Each of these parties require different levels of access as well as have different requirements on data. We describe the design and architecture of SensorSafe [43] and detail the fine-grained context-aware access control policies that it currently supports. In addition, the framework also serves as an implementation base for the two other privacy mechanisms proposed in this paper—controlling data rate by exploiting the trust graph between consumers and providers in Section 4 and feature sharing based privacy control mechanism in Section 5.

The SensorSafe framework as shown in Fig. 1 is a network of users (providers and consumers), a broker, and data stores. The data stores are maintained on private computers, for e.g. at providers' homes, or trustworthy organization's servers creating a virtual machine pool administered locally by their owners. This approach allows providers to store data on their private servers and reduces the risk of data compromise. Providers upload sensory information to their data stores and define privacy rules for the uploaded data. Consumers in turn send their requests for data to the broker, which after appropriate authentication, forwards the request to the providers. After processing of the privacy rules applicable to the requesting consumers, data transfer takes place directly between the data stores and the consumers. This provides for distributed processing and prevents the broker from being a performance bottleneck.

In SensorSafe, the flow of information is controlled by a rule-based sharing mechanism. Each provider maintains a distinct notion of privacy, and rule-based sharing can provide personalized control over shared data. Our framework allows providers to define rules that *allow* or *deny* sharing based on a variety of conditions such as current contexts, locations, timestamps, consumers, and so on. Providers can also choose to transform data (data obfuscation) before sharing. Our rule-processing module, *Privacy Engine*, currently supports generalization of locations and timestamps and abstraction of raw sensor data to

Table 1
Various options for privacy rules.

(a) Conditions and actions

Options		Attributes
Conditions	Consumer	UserName, GroupName
	Location	Label, Region
	Time	Range, Repeat
	Sensor	Sensor Name
	Context	Moving, Not Moving, Still, Walk, Run, Bike, Drive, Stress, Smoke, Conversation
Actions		Allow, Deny, Modify

(b) Example obfuscation options

Context	Options
Location	Coordinates, Street Address, Zipcode, City, State, Country
Time	Hour, Day, Month, Year
Activity	Accelerometer Data, Still/Walk/Run/Bike/Drive, Move/No Move
Stress	ECG/Respiration Data, Stressed/Not Stressed,
Smoking	Respiration Data, Smoking/No Smoking
Conversation	Microphone/Respiration Data, Conversation/No Conversation,
Hiding home	Cloaking, Noise, Rounding

context labels. An important concern in rule-based sharing is that the defined rules could sometimes be at conflict leading to unintentional information leakage. The Privacy Engine also takes into account rule inconsistencies before enforcing them. Additional details of each of these subsystems are discussed in the following sections.

3.1. Privacy rules

Table 1 summarizes the conditions, actions, and data obfuscation options supported for defining privacy rules.

Basic conditions: Using the consumer condition, providers can specify whether a consumer or a group of consumers will be affected by a rule. Providers specify locations by defining a region on a map based user interface. Temporal conditions are defined as continuous time ranges (e.g., from Feb. 2011 to Mar. 2011) or repeated times (e.g., 3–6 pm on every Wednesday). In addition, using the sensor condition, providers can select specific sensor channels in their privacy rules.

Context condition: Providers can also define rules for context information drawn from sensor data. For example, accelerometer data can be used to infer the driving context and respiration sensors can be used to detect conversation episodes [44]. Using contexts as conditions, providers can define rules such as "do not share any data while driving". or "do not share data while in conversation".

Actions: In addition to the allow and the deny rules, providers can transform data to share coarse and abstract information. For example, instead of raw acceleration data, providers can choose to share transportation modes (e.g., still, walk, run, bike, drive) or just abstract it further to binary levels of stationary versus non-stationary.

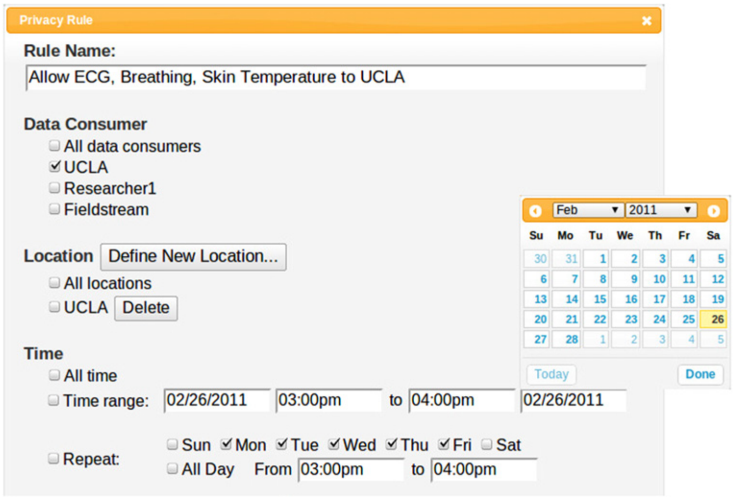
3.2. Rule language

Providers can define privacy rules using our web-based user interface. These rules are then encoded using our rule language, and processed by the Privacy Engine before being translated into database query language. In SensorSafe, we choose to store user rules in the rule language format for its flexibility of translation to any target query language.

We use MongoDB [45] as our database and extend its query language, which currently supports a complete set of conditional and Boolean operators, to include repeated temporal constraints and corresponding actions. As shown in the following example, the repeat time option is motivated by the *cron* [46] time specification utility under Linux operating system. The digits in order represent seconds, minutes, hours, days of week, days of month, months, and years. The action option includes *allow*, *deny*, and *modify* with detailed specifications of what operation should be performed on the data. For example, a rule:

```
[
  { consumer: 'Cathy',
  action: { modify: 'HideHome'} },
  { consumer: 'Cathy',
    location_label: 'Campus',
    repeat_time: '* * 9-18 1-5 * * *',
    context: 'Conversation',
    action: { deny: 'Stress' } },
]
```

means that "Share all data with Cathy but hide home location, and do not share any stress related data while in conversation on campus during weekdays from 9 am to 6 pm".


Privacy Rule ×

Rule Name:
Allow ECG, Breathing, Skin Temperature to UCLA

Data Consumer
- ☐ All data consumers
- ✓ UCLA
- ☐ Researcher1
- ☐ Fieldstream

Location Define New Location...
- ☐ All locations
- ☐ UCLA Delete

Time
- ☐ All time
- ☐ Time range: 02/26/2011 03:00pm to 04:00pm 02/26/2011

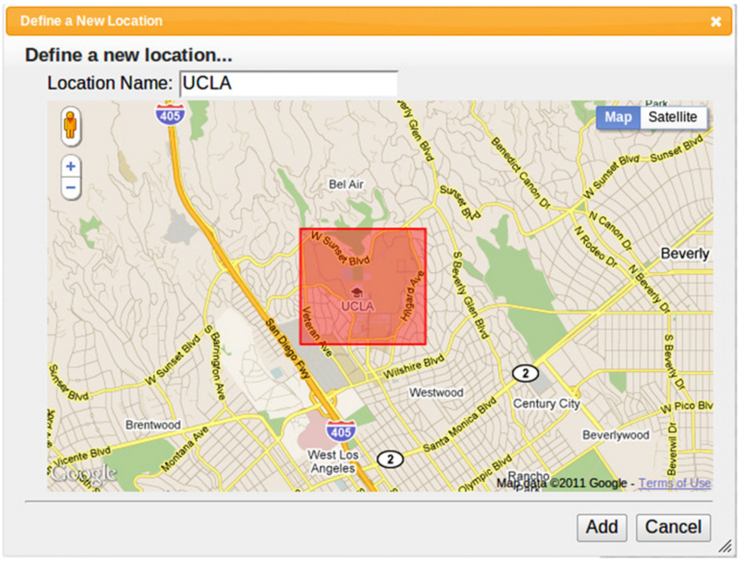
☐ Repeat: ☐ Sun ✓ Mon ✓ Tue ✓ Wed ✓ Thu ✓ Fri ☐ Sat
☐ All Day From 03:00pm to 04:00pm

Feb 2011

Su	Mo	Tu	We	Th	Fr	Sa
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	1	2	3	4	5

Today Done

Fig. 2. Web based user interface for privacy rules.


Define a New Location ×

Define a new location...

Location Name: UCLA

Add Cancel

Fig. 3. Location preferences specified using a map interface.

To simplify privacy rule creation of a user, we provide a web-based user interface as shown in Figs. 2 and 3. It consists of a map, calendars, dialog boxes, and common HTML UI components. Although usability study remains is a planned future work, we believe users will find it easy to use these interfaces owing to familiarity with the UI elements used.

3.3. Conflicting rules

Data from a single sensor can be used to infer multiple contextual information (e.g., a respiration data can be used for stress, conversation, and smoking). Therefore, it is essential to determine the consistency of the user defined privacy rules. For example, if we have a rule denying smoking contexts, respiration data should not be shared even though we have a rule that allows the data (conflicting rules). This is because once the respiration data are provided, smoking can be inferred from the data. The Privacy Engine maintains the relationships between sensors and the various contexts to guarantee that the data release is consistent with the privacy rules. In case of conflicts, providers are notified. Currently, the relationships between contexts and sensors are obtained from domain experts.

3.4. Trust model and feature sharing mechanism

The producers and consumers need to register with the broker to obtain authentication keys. In addition, we envisage that in the future, the broker would also provide providers/consumers with an interface to provide ratings about

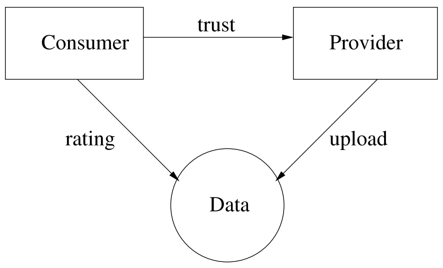


Fig. 4. Computing trust scores from consumer ratings.

consumers/providers after a data transaction. This rating will be used to derive trust scores and support the mechanism described in Section 4. In addition, every data store will locally implement the feature sharing mechanism in Section 5 and transform data before sharing.

4. How much to share: balancing utility and privacy

We consider a problem setting where a single provider wants to determine the resolution (defined later) at which it can share data with multiple consumers while satisfying its privacy and utility requirements. Using resolution as the primitive, the provider specifies different privacy requirements for each consumer. This is the maximum resolution at which it is willing to share data. Each consumer also advertises the minimum resolution that it needs for providing the desired utility. If the provider strives to exactly satisfy every consumer request it could end up violating its own privacy. Similarly, not receiving the appropriate data resolution will result in degraded service utility at the consumer. We want to determine the optimal resolution at which the provider should share data with each consumer to ''closely'' satisfy both privacy and utility constraints.

An observation is that there exists prior trust relationships (defined later) between consumers. This trust network is an important indicator of information flow within a network. Consequently, the decision of *how much* data to share with a particular consumer depends not only on the trust between the provider and consumer, but also on the existing trust network between a consumer and other neighboring consumers [25].

In this section, we start by defining the notion of data resolution, followed by a brief description of trust. We then formulate the trade-off between utility and privacy as a linear program and provide simulation results at the end.

4.1. Resolution

Information quality of data is a function of multiple dimensions such as accuracy, currency and completeness [47]. Depending on the privacy desired, a combination of these dimensions can be appropriately obfuscated. We summarize the effect of these parameters into a single dimensionless number, which we refer to as *resolution* r, and normalize it to values in [0, 1]. The interpretation of r is application and data specific. For location data, r could be accuracy with which the GPS coordinates are provided to location based services [18,43]. An intuitive interpretation for the discrete case could be specifying the exact GPS coordinates versus specifying coarser levels such as building number, street name, city, or state. For images, r could be the fraction of the total number of pixels shared which again relates to the accuracy of the image. For accelerometer data used for activity detection [48], or physiological data such as ECG [13], it could mean the fraction of the total number of samples shared (completeness). For real-time monitoring applications, r could be the tolerable delay (currency). Also, not every data type would retain their utility after obfuscation. Hence, we restrict ourselves to the class of sensory data such as location, accelerometer, images, speech etc. which offers utility even after changes to resolution.

4.2. Modeling trust

An important aspect of any system with multiple data providers and recipients is the modeling and update of trust relationships between its constituents [49,50]. The process of defining and interpreting trust is highly subjective [51], and as a result trust has found diversified uses depending on the application domain. For example, in e-commerce applications [52,53] trust is used as a soft security mechanism. In p2p and social networks it establishes well-knit credible social structures [54] and in sensor networks it is used to assess the quality of information received [55]. In this paper, we follow the definition in [56] and interpret trust as a subjective probability with which an agent assesses that another agent or group of agents will perform a particular action both before he can monitor such action or independent of his monitoring capability.

Similar to interpretation, there are different ways to quantify reputation-based trust [57]. We follow a transaction-based model [55], the underlying idea of which is illustrated in Fig. 4. The provider uploads data, which is rated by the consumers. Trust is then derived as a function of the given ratings. The Beta distribution based trust model [58], due to its flexibility and simplicity is a popular way of quantifying trust and has been used in several trust-based frameworks [59,55]. The beta

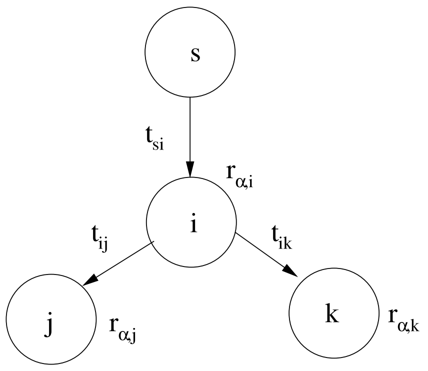


Fig. 5. A trust subgraph. i, j and k are the consumers. t_{ij} and t_{ik} represent trust that i has on j and k. $r_{\alpha,i}$, $r_{\alpha,j}$ and $r_{\alpha,k}$ are resolutions that the provider shares with i, j, and k respectively.

probability density functions are used to combine the positive and negative ratings from consumers for successful and a failed transactions respectively, and derive average trust scores. SensorSafe, in future, would implement an interface for consumers to provide ratings based on the transaction experience with the providers. This would form the basis for deriving the trust scores between the providers and the consumers.

From a privacy perspective, trust has different implications for the data provider and consumer. At the provider, the heterogeneity in trust scores manifests itself as different privacy requirements. If a consumer is not trustworthy, then strict privacy is desired. At the other end, the consumer's trust on a provider is an indicator of the expected quality of information. A trustworthy provider is expected to provide high quality information.

Thus, the notion of trust, which in many ways is the precursor to privacy concerns, is also a unifying entity—describing both the desired obfuscation at the provider and the expected utility at the consumer.

4.2.1. *Trust graph*

Let $T = (V, E)$ be a weighted complete graph with vertex set V representing the producers and consumers in the network and the edge weights representing the trust scores between them. We will refer to the graph T as the trust graph. In case of past transactions between nodes, we compute *direct trust* based on the ratings of those transactions [50]. In the absence of any transaction, we compute *indirect trust* assuming a transitive propagation of trust relationship in the network [54,60]. In case of multiple paths, we conservatively consider the minimum value as the trust score. Thus, every node computes and maintains trust for every other node in the network. Fig. 5 shows a subgraph of T that exists between provider s and consumers i, j and k. The edge weights represent trust scores. Thus, $0 \leq t_{ij} \leq 1$ is how much i trusts j and a higher value indicates greater trust. We interpret t_{ij} as the probability with which i will share data with j. We assume that T is known to the provider.

4.3. *Notation*

Let $\mathcal{R}$ be the set of consumers. The vector $R = \{r_1, r_2, \ldots, r_{|\mathcal{R}|}\}$ is the set of maximum data resolution requested by consumers. Parameter $0 \leq \alpha \leq 1$, is provider specified and represents the tolerance scale to privacy violation. A lower value of α implies higher privacy, and a higher value means allocating each consumer closer to the requested resolution resulting in greater utility. Vector $R^*_\alpha = \{r^*_{\alpha,1}, r^*_{\alpha,2}, \ldots, r^*_{\alpha,|\mathcal{R}|}\}$ is the set of optimal resolutions at which the provider should share data for a particular value of α.

4.4. *Risk of leakage*

Intuitively, risk is an educated guess of the possible loss or damage that could arise out of a particular decision. There are multiple subjective interpretations of risk. In our work, we quantify *risk of disclosure* = *probability of disclosure* × *value of information* where *probability of disclosure* is a prediction that a consumer will share the data with other consumers based on prior experience and *value of information* is the damage sustained by the provider of information due to unauthorized disclosure [61].

For predicting node behavior, we use the edge weights in the trust graph T. For estimating the value of information leaked, we assume a *monotonic* relation between data utility and data resolution. Thus, if $F(r)$ is the utility at resolution r then $r_1 \geq r_2 \Rightarrow F(r_1) \geq F(r_2)$. This is generally true for applications, as better quality data typically yields better results. Thus, the value of information leaked is proportional to the increase in utility at the consumer which was otherwise allocated a lower resolution by the provider.

Let the trust subgraph and the resolution at which the provider shares data with consumers i, j and k be as shown in Fig. 5. To compute the risk that a provider s incurs when it wants to share data with consumer i at resolution $r_{\alpha,i}$ we consider

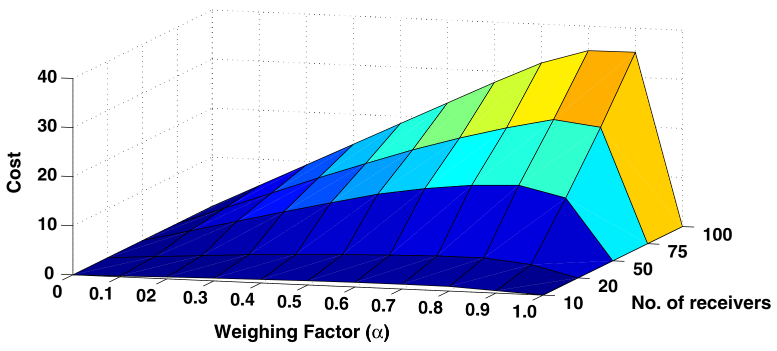


Fig. 6. Shape of the cost function for different number of consumers. Total number of nodes in the network is 200.

the following two cases:

(i) $r_{\alpha,i} \leq r_{\alpha,j}$: risk $= 0$. This is because we assume that information leakage occurs when a consumer obtains information at a resolution higher than that determined by the provider.
(ii) $r_{\alpha,i} > r_{\alpha,j}$: risk $= t_{ij} \times (r_{\alpha,i} - r_{\alpha,j})$.

We compute the risk between pairs i and k in a similar way. Combining the two cases above we define the risk function $f(\cdot)$ for node i as:

$$f(T, R_\alpha, i) = \sum_{\{j \in \mathcal{R}, j \neq i\}} t_{ij} \times [r_{\alpha,i} - r_{\alpha,j}]^+ \tag{2}$$

where $[x]^+ = \max(x, 0)$. Eq. (2) is for one level of direct leakage from a given consumer (i.e., from $s \to i \to j$ for consumer i). It does not account for the higher level cascaded leakages (i.e., from $s \to i \to j \to k$ (two levels) or other higher levels for consumer i). While there exist possibilities of these leakages, as we go down the chain for higher levels, the probability of leakage (which is the product of the trust scores) decreases. Therefore, for reasons of simplicity, we take into account only the direct leakages which, under consistent trust score, are also the most dominant terms of the cascaded series for each consumer.

4.5. Formulation

Using the risk function in Eq. (2) we can define the following numerical optimization problem.

$$\min \sum_{i \in \mathcal{R}} \left(\alpha(r_i - r_{\alpha,i}) + (1-\alpha) \sum_{\{j \in \mathcal{R}, j \neq i\}} t_{ij}[r_{\alpha,i} - r_{\alpha,j}]^+ \right) \tag{3}$$

$$\text{s.t } r_{\alpha,i} \leq r_i \quad \forall i \in \mathcal{R} \tag{4}$$

$$r_{\alpha,i} \geq \min\{r_i | r_i \in R\}. \tag{5}$$

The objective function in Eq. (3) has two parts. The first part weighed by parameter α tries to maximize the utility by allocating a resolution $r_{\alpha,i}$ as close to resolution r_i *sought* by the consumer. We refer to the difference between the sum of the allocated resolution and the requested resolution as the *fidelity cost*. Thus, higher fidelity cost yields lower utility. The second part weighed by $1 - \alpha$ is the risk function derived in Eq. (2). Constraint (4) ensures that the allocated resolution does not exceed the maximum resolution required by the consumer. Constraint (5) ensures that the allocated resolution is at least as high as the minimum of the application specific resolutions requested by the consumers. This problem can be easily cast as a Linear Programming problem [62] and the optimal solution R^*_α found using standard LP solvers. R^*_α contains the sharing constraint for each consumer.

4.6. Simulation results

We summarize our simulation results in this section. We used a connected graph of 200 nodes as our network. We randomly choose a provider node. The trust values for the graph T are chosen uniformly from the interval [0, 1]. The number of consumers in the set $\mathcal{R}$ are chosen from {10, 20, 50, 75, 100}. The sets are incrementally generated implying that the set of 20 consumers, includes the set of previously chosen 10 consumers and so on. The R vector, for the selected consumers, is generated by choosing values uniformly from the interval [0, 1]. The solution is scalable to a larger network of nodes and number of consumers.

Variation in cost function.

The variation in the optimal objective function value for different number of consumers is shown in Fig. 6. Each plot is generated by varying α in steps of 0.1 in the interval [0, 1]. The cost function has a unimodal distribution and scales

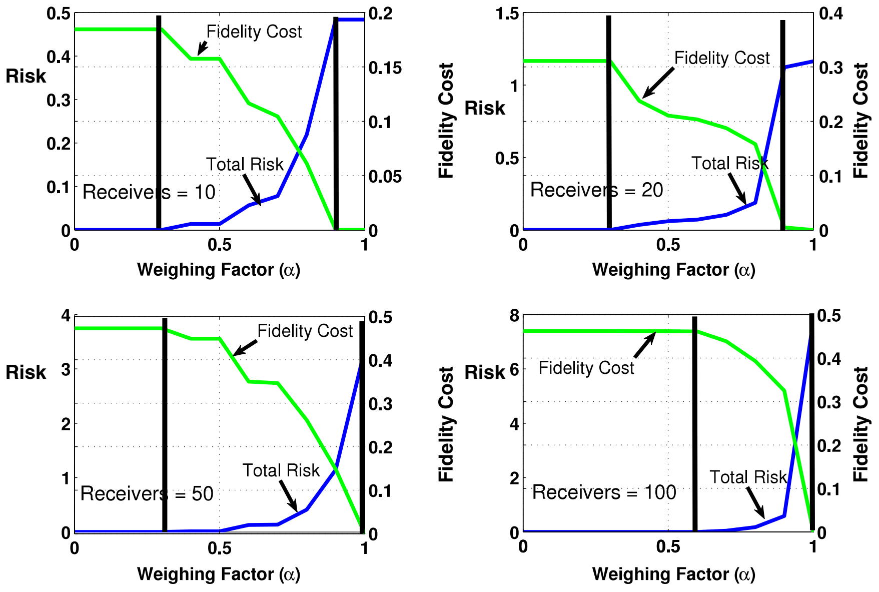


Fig. 7. Effect of α on total risk and fidelity cost for different number of consumers. The total number of nodes in the network is 200.

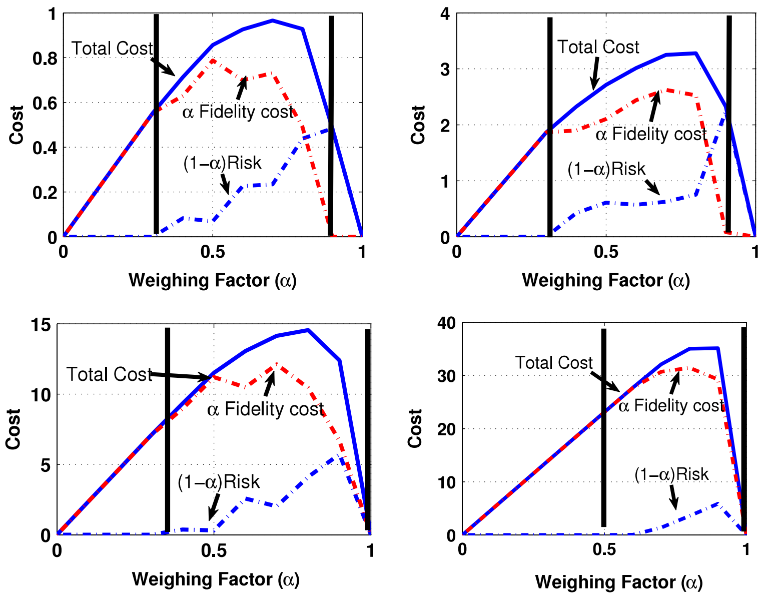


Fig. 8. Contribution of risk and fidelity to the total cost for different number of consumers {10, 20, 50, 100}. The total number of nodes in the network is 200.

with increase in the number of consumers. This is because of the additive increase to the individual terms of the objective function.

Variation between α, Application Fidelity and Total Risk: Fig. 7 shows how average risk and fidelity cost changes with α for different number of consumers. For smaller values of α the risk of leakage is minimized whereas for larger values utility is preferred over leakage.

Contribution to the total cost: The contribution of fidelity cost and risk of leakage toward the total cost (Eq. (3)) is shown in Fig. 8 for different number of consumers. As shown in the figure by the black lines, the domain of α values can be partitioned into three distinct regions, and the provider could choose to operate in any of the regions. In the first region, the provider chooses to minimize risk over application fidelity. In the third region, the provider prefers application fidelity over risk. However, in the middle region, both fidelity cost and risk contribute to the total cost. The provider could choose α in this range to balance non-zero risk and non-zero application fidelity. This middle region is important because trade-off between risk and application fidelity is only possible in this region. Depending on the number of consumers, the boundary values of α for these three regions could vary.

5. What to share: protecting behavioral privacy

The final step of the solution framework provided by SensorSafe is to determine what data to share with the consumer such that inference violations could be prevented. In this section, we start by formalizing the notion of inference violation. We then propose a compressive sensing based obfuscation scheme for selectively perturbing features extracted from data while satisfying the utility and privacy objectives. Finally, we present a case study using a simple example scenario to show the proof-of-concept of the proposed approach.

5.1. Formalization

Let U be the universe of computable inference functions. A function $g(\cdot) \in U$, takes as input data $\mathbf{x} \in \mathbb{R}^n$ and computes $\mathbf{y} \in \mathbb{R}^n$, i.e. $g(\mathbf{x}) = \mathbf{y}$. Let O be any obfuscation function applied on the input $\mathbf{x}$ to produce $\tilde{\mathbf{x}}$, i.e. $\tilde{\mathbf{x}} = O(\mathbf{x})$. The error in the computation of inference function $g(\cdot)$ using actual and obfuscated data is given by $\mathrm{err}(g) = |g(\mathbf{x}) - g(\tilde{\mathbf{x}})|$. We define numbers $s, l \in \mathbb{R}$ such that $l \gg s$. We define a "black" list B and a "white" list W such that $W, B \subseteq U$, and $W \cap B = \emptyset$. Thus, given the subsets W and B, we want to *design an obfuscation function O*, such that:

$$\forall w \in W \quad \mathrm{err}(w) \leq s \tag{6}$$

$$\forall b \in B \quad \mathrm{err}(b) \geq l. \tag{7}$$

Eq. (6) corresponds to the white-listed inference functions which should be computed with low error or high accuracy. The number s can be interpreted as the maximum tolerance to utility loss. Similarly in Eq. (7), using the same obfuscated data $(O(\mathbf{x}))$ the black-listed functions should not be accurately computable. l is the minimum privacy constraint that should always be satisfied for functions in subset B. Set $G = U \setminus \{B \cup W\}$ are functions which are not classified into either of the two lists. A "privacy first" approach would try to protect against this "gray" list of excluded functions as well. The privacy scheme O should satisfy the constraints in Eqs. (6) and (7) simultaneously.

5.2. Sharing features

For a multi-dimensional data stream $\mathbf{x} \in \mathbb{R}^n$, an inference can be modeled as a function computed over a subset of the data dimensions. Let us consider two inference functions $w()$ and $b()$, that are part of the white and black list respectively. That is, the user is willing to share the inference derived by computing $\mathbf{y}_w = w(\mathbf{x})$, but would like to conceal the inference derived by computing $\mathbf{y}_b = b(\mathbf{x})$. Notice that a core assumption here is that the function $w()$ is non-invertible, because if $w()$ was invertible, a malicious user with access to $\mathbf{y}_w$ could compute the black list inference through a trivial intermediate step: $y_b = b(w^{-1}(y_w))$. In practice, the assumption is both reasonable and desirable. When true, the assumption implies that the inference function is *dimensionality reducing*. This is a significant implication for our work and is a cornerstone of our approach. Computing an inference is typically achieved by first projecting the data onto a feature space that "sparsifies" it in that domain. The domain chosen for sparsification depends not only on signal characteristics but also on the inference being evaluated. The projection step is followed by an identification and elimination step that retains only the dominant features in the sparse domain, hence reducing the signal's dimensionality. One may view, therefore, each inference function in the white and black lists as defining a distinct basis in which the signal is sparse.

We have considered compressive sensing [27,63] for determining the level of data obfuscation. Compressive sensing has been widely used for near accurate reconstruction of sparse signals using sampling rates much lower than traditional Nyquist sampling. Compressive Sensing [27] also provides tools to provide guarantees for recovering a sparse signal given a projection matrix that delivers a set of transformed data measurements. By applying the fact that each inference can be viewed effectively as a sparsifying basis, we can utilize existing theoretical results to design appropriate transformations that meet our requirements. An example of this is shown in Figs. 9 and 10.

5.3. Case study

Consider an arbitrary signal, shown in Fig. 9 in the time domain and its representations in the discrete cosine transform (DCT) [64] and discrete wavelet transform (DWT) with Haar wavelets [65]. The DCT domain is known to pick smooth undulations in the signal while the Haar transform is known to identify edges. While this is a simple example, it could easily be extended to more sophisticated linear classification algorithms such as support vector machines, where the feature basis and the SVM parameters are used collectively to define the inference function. Fig. 10 is an obfuscated version of the raw signal arrived at by applying a "confusion" function that explicitly preserves the information in the dominant component (feature) in the DCT domain while deteriorating the quality of the corresponding feature in the DWT domain. To achieve this obfuscation, we keep the dominant component in the DCT domain and discard the rest of the signal. We then, transform this signal to the DWT domain and after setting the dominant feature close to zero, take an inverse DWT of the normalized signal. The signal in the time domain is shared. Fig. 10 illustrates that the obfuscation, while degrading the sparsity of the DCT domain signal, does not obscure the most significant inference feature. On the other hand, the coefficients in the DWT and time domain are distorted to a large extent.

From compressive sensing results, it is known that the degradation in reconstruction accuracy varies monotonically with the number of samples as well as the per-sample error within a known probabilistic bound [63]. This implies that,

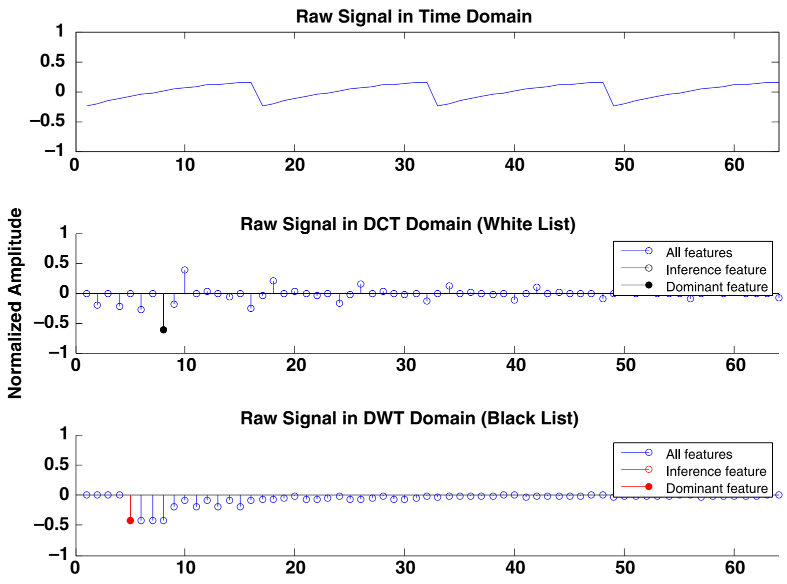


Fig. 9. An example of multi-basis privacy: raw data represented in the sensing (time) domain and in domains that are in the white list (DCT) and black list (DWT).

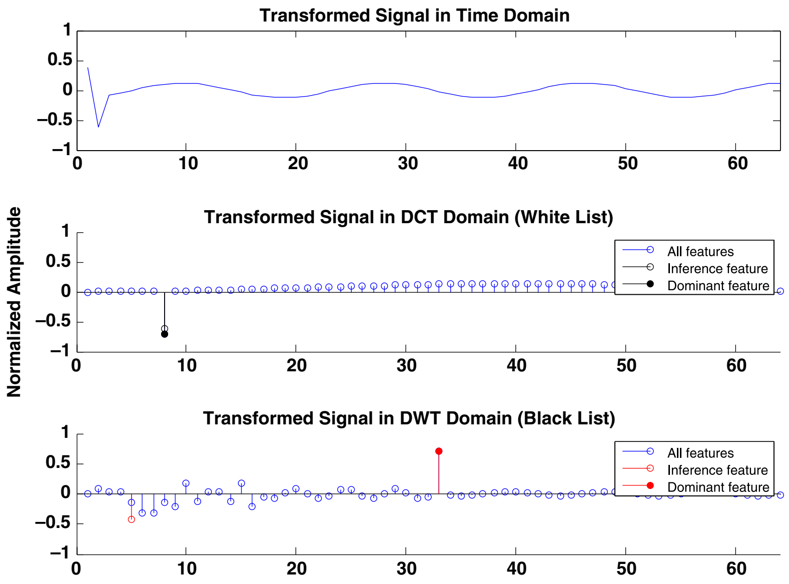


Fig. 10. An example of multi-basis privacy: transformed data represented in the sensing (time) domain and in domains that are in the white list (DCT) and black list (DWT).

when using compressive sensing, the highest magnitude coefficients are recovered first, even when the number of sample measurements is insufficient for exact recovery. We exploit this property to restrict the number of measurements such that only the dominant features in the white listed inference bases can be recovered but not the inference features in the black list DWT domain with high probability. To achieve this, we precisely calibrate the distribution and amount of noise added for obfuscating the values corresponding to the dimensions in the function $b()$. In addition, compressive sensing requires us to work in domains where the signal has a sparse representation. This allows us to reduce the magnitude of the overall noise that we need to add to the system allowing us to preserve utility while maintaining privacy.

Obviously, the compressive sensing approach fails for some classes of white and black list inferences. For example, if a black list inference can be derived from a white listed one, or if the black list inference is just the identity. The key problem to be addressed within this framework is that of defining an appropriate distance metric between inference bases to provide the probabilistic guarantees in Eqs. (6) and (7) to the data provider based on their choice of white and black listed behavioral inferences and parameters s and l respectively. We look to extend this approach, over multiple different inference functions the with varying privacy and utility concerns and under different interactions between the data dimensions.

6. Conclusion

In this paper, we recognize and present a new privacy issue that stems from sharing personal data streams—behavioral privacy. Behavioral privacy arises when unintended inferences about lifestyle patterns are mined from shared data and we showed how existing solutions to prevent identity privacy and privacy due to incomplete reconstruction become inadequate as some inferences can be drawn from partial data too.

As a first step toward addressing these challenges, we described the design and architecture of SensorSafe. SensorSafe takes into account multiple facets of the privacy problem and implements mechanisms to handle each of them. It starts with providing a flexible user interface for fine grained and context dependent access control. Then, it takes into account the trust structure between providers and consumers to determine the rate of information flow, and finally proposes a mechanism which uses inferences as primitives together with compressive sensing to simultaneously achieve the utility guarantee for the intended or white-listed inferences while simultaneously preventing the unintended or black-listed inferences.

Acknowledgments

This research was sponsored by the US Army Research Laboratory and the UK Ministry of Defense under Agreement Number W911NF-06-3-0001 and by the NSF under award #0910706. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defense or the UK Government or the NSF. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] S. Goldwasser, S. Micali, Probabilistic encryption and how to play mental poker keeping secret all partial information, in: Proc. of the Fourteenth Annual ACM Symposium on Theory of Computing, STOC, 1982, pp. 365–377.
[2] C. Dwork, Differential privacy, in: Int. Colloquium on Automata, Languages and Programming, ICALP, 2006, pp. 1–12.
[3] T. Li, N. Li, On the tradeoff between privacy and utility in data publishing, in: Proc. of the 15th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, KDD, 2009, pp. 517–526.
[4] L. Sankar, S. Rajagopalan, V. Poor, A theory of utility and privacy of data sources, in: IEEE Int. Symposium on Information Theory Proc., ISIT, 2010, pp. 2642–2646.
[5] L. Sweeney, *k*-anonymity: a model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based 10 (2002) 557–570.
[6] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, *L*-diversity: privacy beyond *k*-anonymity, ACM Transactions on Knowledge Discovery from Data 1 (2007).
[7] N. Li, T. Li, S. Venkatasubramanian, *t*-closeness: privacy beyond *k*-anonymity and *l*-diversity, in: IEEE Int. Conference on Data Engineering, ICDE, 2007, pp. 106–115.
[8] M. Demirbas, M.A. Bayir, C.G. Akcora, Y.S. Yilmaz, H. Ferhatosmanoglu, Crowd-sourced sensing and collaboration using twitter, in: Proc. of the 2010 IEEE Int. Symposium on a World of Wireless, Mobile and Multimedia Networks, WOWMOM, 2010, pp. 1–9.
[9] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, M. Corner, Mcrowd: a platform for mobile crowdsourcing, in: Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys., 2009, pp. 347–348.
[10] http://whatsinvasive.com/.
[11] J. Sriram, M. Shin, D. Kotz, A. Rajan, M. Sastry, M. Yarvis, Challenges in data quality assurance in pervasive health monitoring systems, Future of Trust in Computing (2009).
[12] D. Kotz, S. Avancha, 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, 2009, pp. 1–12.
[13] E. Ertin, N. Stohs, S. Kumar, A. Raij, M. al'Absi, S. Shah, Autosense: unobtrusively wearable sensor suite for inferring the onset, causality, and consequences of stress in the field, in: Proc. of the 9th ACM Conference on Embedded Networked Sensor Systems, SenSys., 2011, pp. 274–287.
[14] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, P. Boda, Peir, the personal environmental impact report, as a platform for participatory sensing systems research, in: Proc. of the 7th Int. Conference on Mobile Systems, Applications, and Services, MobiSys, 2009, pp. 55–68.
[15] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, S. Madden, Cartel: a distributed mobile sensor computing system, in: Proc. of the 4th Int. Conference on Embedded Networked Sensor Systems, SenSys, 2006, pp. 125–138.
[16] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, M. Srivastava, Biketastic: sensing and mapping for better biking, in: Proc. of the 28th Int. Conference on Human Factors in Computing Systems, CHI, 2010, pp. 1817–1820.
[17] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, D. Irwin, Private memoirs of a smart meter, in: Proc. of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, BuildSys., 2010, pp. 61–66.
[18] J. Krumm, A survey of computational location privacy, Personal and Ubiquitous Computing 13 (6) (2009) 391–399.
[19] http://www.h-online.com/security/news/item/Smart-meters-reveal-TV-viewing-habits-1346385.html.
[20] Y. Kim, T. Schmid, M.B. Srivastava, Y. Wang, Challenges in resource monitoring for residential spaces, in: Proc. of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, BuildSys., 2009, pp. 1–6.
[21] A. Raij, A. Ghosh, S. Kumar, M. Srivastava, Privacy risks emerging from the adoption of innocuous wearable sensors in the mobile environment, in: Proc. of the 2011 Annual Conference on Human Factors in Computing Systems, CHI, 2011, pp. 11–20.
[22] S. Hansell, AOL removes search data on vast group of web users, New York Times, 2006, URL: http://query.nytimes.com/gst/fullpage.html?res=9504E5D81E3FF93BA3575BC0A9609C8B63.
[23] A. Narayanan, V. Shmatikov, Robust de-anonymization of large sparse datasets, in: Proc. of the 2008 IEEE Symposium on Security and Privacy, 2008, pp. 111–125.
[24] A. Harmon, Where'd you go with my DNA, New York Times, 2010, URL: http://www.nytimes.com/2010/04/25/weekinreview/25harmon.html?pagewanted=all.

[25] S. Chakraborty, H. Choi, M.B. Srivastava, Demystifying privacy in sensory data: a QoL based approach, in: IEEE Int. Conference on Pervasive Computing and Communications Workshops, 2011, pp. 38–43.
[26] P. Ohm, Broken promises of privacy: responding to the surprising failure of anonymization, in: Social Science Research Network Working Paper, 2009.
[27] E. Candes, T. Tao, Decoding by linear programming, IEEE Transactions on Information Theory 51 (2005) 4203–4215.
[28] B.C.M. Fung, K. Wang, R. Chen, P.S. Yu, Privacy-preserving data publishing: a survey of recent developments, ACM Computing Surveys 42 (2010) 14:1–14:53.
[29] A. Narayanan, V. Shmatikov, Myths and fallacies of "personally identifiable information", Communications of the ACM 53 (2010) 24–26.
[30] R.K. Ganti, N. Pham, Y.-E. Tsai, T.F. Abdelzaher, Poolview: stream privacy for grassroots participatory sensing, in: Proc. of the 6th ACM Conference on Embedded Network Densor Dystems, SenSys., 2008, pp. 281–294.
[31] H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, J. Han, Privacy-aware regression modeling of participatory sensing data, in: Proc. of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys., 2010, pp. 99–112.
[32] J. Shi, R. Zhang, L. Yunzhong, Y. Zhang, Privacy-preserving data aggregation in people-centric urban sensing systems, in: Proc. IEEE Infocom, 2010.
[33] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, Theory of Cryptography 3876 (2006).
[34] R. Sarathy, K. Muralidhar, Evaluating laplace noise addition to satisfy differential privacy for numeric data, Transactions on Data Privacy 4 (2011) 1–17.
[35] R. Cáceres, L. Cox, H. Lim, A. Shakimov, A. Varshavsky, Virtual individual servers as privacy-preserving proxies for mobile devices, in: Proc. of the 1st ACM workshop on Networking, Systems, and Applications for Mobile Handhelds, MobiHeld, 2009, pp. 37–42.
[36] S.-W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S.K. Teh, R. Chu, B. Dodson, M.S. Lam, PrPL: a decentralized social networking infrastructure, in: Proc. of the 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond, MCS, 2010, pp. 8:1–8:8.
[37] A. Tootoonchian, S. Saroiu, Y. Ganjali, A. Wolman, Lockr: better privacy for social networks, in: Proc. of the 5th Int. Conference on Emerging Networking Experiments and Technologies, 2009.
[38] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, D. Starin, Persona: an online social network with user-defined privacy, in: Proc. of the ACM SIGCOMM 2009 Conference on Data Communication, 2009, pp. 135–146.
[39] E. Toch, J. Cranshaw, P. Hankes-Drielsma, J. Springfield, P.G. Kelley, L. Cranor, J. Hong, N. Sadeh, Locaccino: a privacy-centric location sharing application, in: Proc. of the 12th ACM Int. Conference Adjunct Papers on Ubiquitous Computing, Ubicomp., 2010, pp. 381–382.
[40] M. Mun, S. Hao, N. Mishra, K. Shilton, J. Burke, D. Estrin, M. Hansen, R. Govindan, Personal data vaults: a locus of control for personal data streams, in: Proc. of the 6th Int. Conference, Co-NEXT, 2010, pp. 17:1–17:12.
[41] D. Boneh, A. Sahai, B. Waters, Functional encryption: definitions and challenges, in: Proc. of the 8th Conference on Theory of Cryptography, TCC, 2011, pp. 253–273.
[42] C. Gentry, Fully homomorphic encryption using ideal lattices, in: Proc. of the 41st Annual ACM Symposium on Theory of Computing, STOC, 2009, pp. 169–178.
[43] H. Choi, S. Chakraborty, Z.M. Charbiwala, M.B. Srivastava, Sensorsafe: a framework for privacy-preserving management of personal sensory information, in: Proc. of the 8th VLDB International Conference on Secure Data Management, SDM, 2011, pp. 85–100.
[44] M. Rahman, A.A. Ali, K. Plarre, M. Absi, E. Ertin, S. Kumar, mConverse: inferring conversation episodes from respiratory measurements collected in the field categories and subject descriptors, work.
[45] MongoDB. URL: http://www.mongodb.org/.
[46] cron, Wikipedia. URL: http://en.wikipedia.org/wiki/Cron.
[47] C. Bisdikian, On sensor sampling and quality of information: a starting point, in: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOMW, 2007, pp. 279–284.
[48] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, Using mobile phones to determine transportation modes, ACM Transactions on Sensor Networks 6 (2) (2010) 13:1–13:27.
[49] H. Li, M. Singhal, Trust management in distributed systems, Computer 40 (2007) 45–53.
[50] K. Govindan, P. Mohapatra, Trust computations and trust dynamics in mobile adhoc networks: a survey, IEEE Communications Surveys & Tutorials PP (99) (2011) 1–20.
[51] A. Jøsang, R. Hayward, S. Pope, Trust network analysis with subjective logic, in: Proc. of the 29th Australasian Computer Science Conference—Volume 48, ACSC, 2006, pp. 85–94.
[52] L. Xiong, L. Liu, A reputation-based trust model for peer-to-peer ecommerce communities, in: Proc. of the 4th ACM Conference on Electronic Commerce, EC, 2003, pp. 228–229.
[53] S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina, The eigentrust algorithm for reputation management in p2p networks, in: Proc. of the 12th Int. Conference on World Wide Web, WWW, 2003, pp. 640–651.
[54] F.E. Walter, S. Battiston, F. Schweitzer, Personalised and dynamic trust in social networks, in: Proc. of the Third ACM Conference on Recommender Systems, RecSys., 2009, pp. 197–204.
[55] S. Ganeriwal, L.K. Balzano, M.B. Srivastava, Reputation-based framework for high integrity sensor networks, ACM Transactions on Sensor Networks 4 (2008) 15:1–15:37.
[56] D. Gambetta, Can we trust trust? in: Trust Making and Breaking Cooperative Relations, 2000.
[57] Q. Zhang, T. Yu, A classification scheme for trust functions in reputation-based trust management, in: ISWC Workshop on Trust, Security, and Reputation on the Semantic Web., 2004.
[58] A. Jøsang, Trust and reputation systems, in: Foundations of Security Analysis and Design, 2007.
[59] S. Buchegger, J.Y. Le Boudec, A robust reputation system for p2p and mobile ad-hoc networks, in: Proc. of the Second Workshop on the Economics of Peer-to-Peer Systems, 2004.
[60] C.-W. Hang, Y. Wang, M.P. Singh, Operators for propagating trust and their evaluation in social networks, in: Proc. of The 8th Int. Conference on Autonomous Agents and Multiagent Systems—Volume 2, AAMAS '09, 2009, pp. 1025–1032.
[61] P.-C. Cheng, P. Rohatgi, C. Keser, P.A. Karger, G.M. Wagner, A.S. Reninger, Fuzzy multi-level security: an experiment on quantified risk-adaptive access control, in: Proc. of the 2007 IEEE Symposium on Security and Privacy, SP, 2007, pp. 222–230.
[62] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[63] Z. Charbiwala, S. Chakraborty, S. Zahedi, Y. Kim, M.B. Srivastava, T. He, C. Bisdikian, Compressive oversampling for robust data transmission in sensor networks, in: Proc. IEEE Infocom., 2010, pp. 1–9.
[64] N. Ahmed, T. Natarajan, K.R. Rao, Discrete cosine transfom, IEEE Transactions on Computers 23 (1974) 90–93.
[65] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1989) 674–693.