Enabling Privacy Through Transparency

Oshani Seneviratne
Decentralized Information Group
MIT CSAIL
Email: oshani@csail.mit.edu

Lalana Kagal
Decentralized Information Group
MIT CSAIL
Email: lkagal@csail.mit.edu

***Abstract*—Many access control systems, particularly those utilized in hospital environments, exercise optimistic security, because preventing access to information may have undesirable consequences. However, in the wrong hands, these over-broad permissions may result in privacy violations. To circumvent this issue, we have developed Privacy Enabling Transparent Systems (PETS) that makes transparency a key component in systems architectures. PETS is built on open web standards and introduces the Provenance Tracking Network (PTN), an open global trusted network of peer servers, to the traditional web stack. Websites that conform to the architecture communicate information about transactions for any sensitive data items with the PTN. These usage logs are stored in a decentralized manner and can later be queried to check compliance with individual usage restrictions that assert no unauthorized data transfer or usage has taken place. PETS enables data consumers to be transparent with regard to data usages and determine if there has been privacy violations after the fact. We conducted a user study on a healthcare data application built using PETS to see if transparency on access and usage data satisfies expectations of user privacy.**

Keywords*—*Privacy, Transparency, User Choice.

I. Introduction

Many web based systems such as social networking websites, web-accessible health care records, personal tax report creation websites, personalized search and other web-based information systems offer a variety of ways for netizens to engage socially, economically and intellectually with friends and strangers. These systems enable users to enter, use, and transfer sensitive information. There is an implicit trust by the users that the mechanics behind these web systems and other users will not misuse the personal information they provide to the system. In certain domains such as healthcare or finance, the information usage is fairly complex and/or unpredictable that the user may not be completely aware about what is happening with the data, and the potential privacy implications of the data misuses. On the other hand, if we were to make the data strictly private, we could be throwing the baby out with the bath water! There is tremendous good from users sharing the right information with the right people in the right ways: scientists can use data in unexpected ways and discover groundbreaking results that can cure diseases; volunteers can crowdsource to find solutions that may take a considerable time, effort and money otherwise, and etc.

Staddon et al. have shown that the ability to copy, collect, aggregate information in large scale systems and the ease with which it is possible to infer sensitive information in them using publicly available data often results in adverse consequences for users [1]. Access control and encryption mechanisms alone have been proven to be ineffective at addressing modern, web-scale privacy problems such as information leakages from large scale analytics resulting in a wide variety of re-identification attacks and data-misuses [2]. Weitzner et al. have thus introduced the notion of 'Information Accountability' where appropriate use of the data can be determined after the fact from audit logs [3]. It was also shown that in systems that support health care decisions or military information systems where the safety of an individual or a community is at risk, foregoing access control mechanisms and getting to the correct information fast through accountable mechanisms is arguably the better alternative compared to upfront preventive measures [4]. This paper builds on these ideas and presents an implementation of a transparent and accountable system used to provide a better outlook on user privacy.

II. Contributions and Outline

We have previously shown how we can address data reuse issues at a protocol level by augmenting the Web with accountability through HTTPA (HTTP with Accountability) [5]. HTTPA uses headers to transmit usage restrictions between web servers and clients, creates an audit log every time a resource access happens via any of the HTTP verbs, i.e. GET, POST, PUT, etc, and these audit logs can later be retrieved for compliance checks [6]. Continuing on these early work, our primary claim in this paper is, that enabling transparency in the standard web applications stack is a necessity in order to assert data ownership, and privacy of users. We present the design and implementation of a system that has support for 'break glass' scenarios where information can be accessed when necessary, but logs all activity for compliance checks afterwards.

We explain the architecture for Privacy Enabling Transparent Systems (PETS) in Section III highlighting some of its salient features, and discuss how this architecture makes the information flows transparent to the data subject in Section IV. We then motivate the application of this architecture using a use case from the healthcare domain in Section V. We outline the methodology and results of the evaluation of a reference PETS implementation called 'Transparent Health' in Section VI. Related work is presented in Section VII. Future work and conclusions from this work are presented in Sections VIII and IX respectively.

III. Implementing Privacy Enabling Transparent Systems (PETS)

We define PETS to have the following minimal characteristics: (1) Agents, i.e. both users and bots, have a unique identity

mechanism. (2) Sensitive data items have usage restrictions associated with them as defined by the data subject or the data provider. (3) Systems interoperate with respect to data, actions, and agents. (4) Usage logs are tamper-evident, stored separately from the data, and provide non-repudiable evidence of the actions by the agents. (5) Users can efficiently retrieve the logs, check them and take action if there are any data misuses.With these requirements in mind, we have designed an end-to-end PETS architecture as shown in Figure 1.

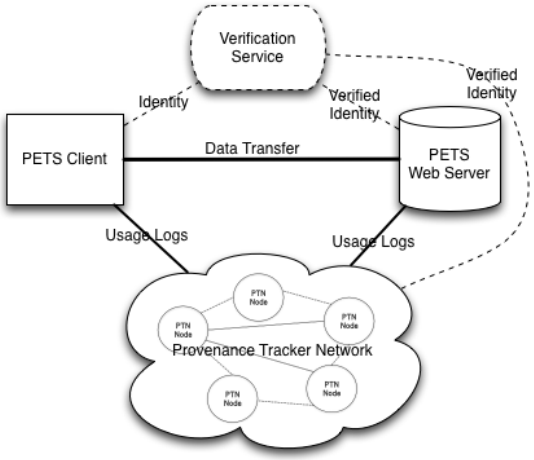

Fig. 1. Building Blocks of PETS: the traditional client server architecture is augmented with provenance information with the help of PTN and the optional Verification Service.

A. Provenance Tracker Network (PTN)

This is a decentralized network of peer servers that maintain the usage logs. No single entity can exercise ownership over the entire collection of log records. We encrypt the usage logs such that only the owner or the data subject of the sensitive data item included in the log record will be able access it. The PTN is implemented using a distributed hash table (DHT), thus, by design the system is fault tolerant, and the lookups are fairly efficient. DHTs support the simple put/get interface from traditional hash tables, but also offer increased capacity and availability by partitioning the key space across a set of participating nodes in a network. By enabling the ownership of log records to be held by a collection of peer nodes rather than a single centralized server ensures that the provenance logs cannot be tampered with. Checksums of the log records from different peers are compared periodically and the peers that are known to tamper the records rejected for integrity.

Our PTN implementation is motivated by OpenDHT [7], which was fairly limited in its programming interface, as it only has **unauthenticated** support for `get` and `put` methods. Also, since it was designed to be a public DHT service that can be used by untrusting services and clients, OpenDHT's storage mechanism does not persist the records. We extended the DHT overlay used in OpenDHT and added the following: (1) The ability to add, update, and retrieve authenticated records with the help of the Authentication, Update and Audit Processors and (2) A persistent storage mechanism using the Log Store. The PTN only stores the log records of the data, and not the actual data items.

Any web application that needs to interface with the PTN can use the PTN wrapper interface shown in Figure 2. We introduce the concepts 'Agents', 'Sensitive Data' and 'Processes' within the wrapper interface that can be configured by the application developer. We have contributed two reference implementations in the Django python web framework and in node.js as middleware modules[1].

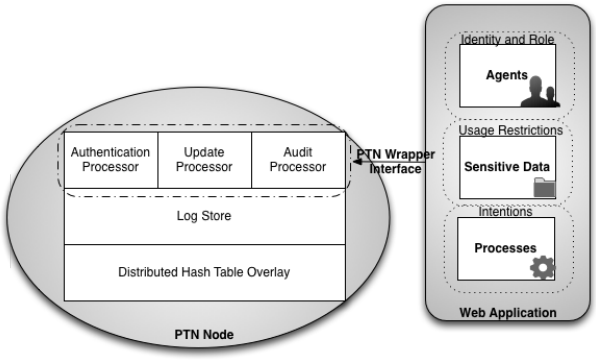

Fig. 2. PTN and a PETS application interact through the PTN wrapper interface.

B. Verification Service

Every agent must have a unique identifier, so that the agent can be identified within the system, as well as in the PTN to ascertain who accessed, used, transferred data in case of compliance checks after-the-fact. Traditional username and password mechanisms do not suffice in our decentralized architecture, as the agents may be acting in different username/password mechanisms. Therefore, a PETS applications may use a 'Verification Agent' to delegate authentication.

We used the Semantic Web based approach for handling global identity using the WebID access control delegation as defined in [8]. A WebID is a URI that refers to an agent, when dereferenced, identify the agent that it represents. The WebID protocol enables global identification of agents using asymmetric cryptography. The origin server, the server where the WebID is hosted, controls the identity of the agent. When an agent needs to authenticate himself to the PTN, the Verification agent can be delegated to do the authentication of the user. The browser based provenance management client will prove the possession of or access to a private key, whose corresponding public key is tightly bound to the WebID that is being authenticated. The private key is associated with an X.509 certificate on the user's computer, and the public key is associated with the agent's WebID profile. We also support a more mainstream authentication approach with the OAuth 2.0 protocol [9] where the agents can use Google OAuth services as the Verification Service. We also plan to add support for other OAuth relying parties in the future.

C. PETS Client and Server

Figure 3 indicates the interactions between the client and the server that involves the client requesting a sensitive data item from the server. After authentication, the client must send

[1]The source code for these libraries are available at https://github.com/mit-dig/httpa.

acknowledgement of the usage restriction terms associated with the data item and also specify the intentions of the request. Only when the identity has been verified and the acknowledgement with the usage intentions has received, the data provider will provide access to the data item. The usage log of this activity will be generated by the data provider's web application, and updated in the PTN.

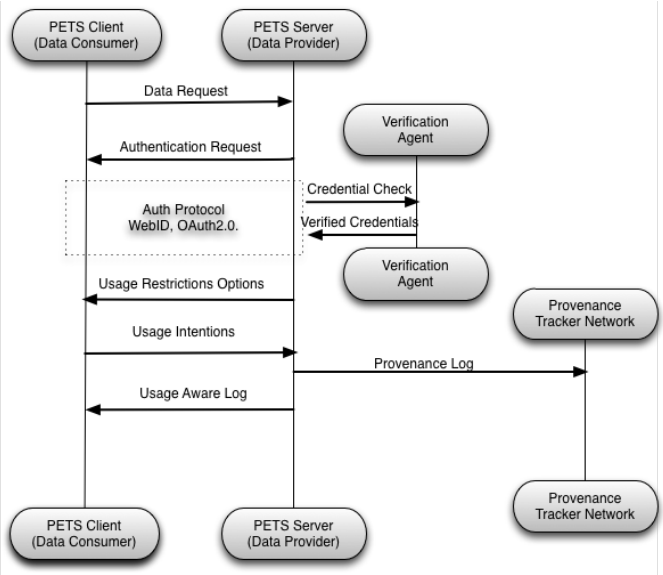

Fig. 3. Interactions in PETS when requesting sensitive data

IV. Making Information Flows Transparent to the Data Subject

A. Annotating Sensitive Data

There are complex data interactions at play in many web information systems. These data interactions include the agents and processes that consume the data, as well as the usage restrictions and intentions imposed on the data. A potential sensitive data item can be marked as such by the application developer on the PETS server. If the sensitive information entails a collection of data items, for example the data field for 'medical conditions', all of the values for 'medical conditions' will be marked sensitive. The decision to leave the classification to the application developer was intentional, as we do not want to inundate the user with the choice as to what constitutes sensitive or not, and any usage restrictions that apply on them. Every time an agent in the system accesses, updates and transfers the sensitive data item through a process, a usage log is created in the PTN by the PETS. Similar to the classification of sensitive data in the system, the definition of the usage restrictions is left entirely to the web application developer. Some of the suggested privacy vocabularies include: Respect My Privacy and Privacy Preference Ontology. The PTN wrapper interface provides methods to communicate these defined usage restrictions on these sensitive data items between the web application and the PTN.

B. Creating Usage Logs for Sensitive Data

The usage logs contain the triple consisting of: a key (k), a value (v), and the hash of a chosen secret up to 40 bytes in length (H). k should be a 160-bit value and the v can be variable-length and there is currently no restriction on the size. All the usage log entries to the PTN are persisted in a datastore at each peer node in the PTN. The usage logs are designed to be immutable except by the owner of the log record. When a usage log is 'put' in the PTN, it is encrypted using the owner's private key K_S, i.e. $\sigma = \{H(k,v)\}_{K_S}$. A 'get' in the PTN should specify both k and K_P, and returns only values that match both k and K_P. The two primary operations 'get' and 'put' in the PTN are summarized in Table 1.

Operation	Returns	Functionality
put(k, v, K_P, σ)	success	Write (k, v), K_P, and $\sigma = \{H(k,v)\}_{K_S}$
get(k, K_P)	{v, σ}	Read v stored under (k, K_P)

TABLE I. Operations for Usage Logs on the PTN

When creating a usage log, first the PTN wrapper attempts a 'get' on the same key that is represented by the URI of the sensitive data item. If there is an existing key in the PTN, those corresponding usage logs are retrieved. As in any DHT implementation, our PTN is also susceptible to churn. Therefore, some PTN peers might have an older version of the provenance log for a given data item because it went offline when a previous update was received. Therefore, our algorithm checks the values of all the keys retrieved within an allotted time. This time can be configured by the application developer, and defaults to 5 seconds. We use the values of the `prov:atTime`, as defined in the provenance ontology [10], in the usage logs retrieved for the given key to determine the newest entry. Once such a value was determined, the new triples are appended to that record. This new $\langle k,v \rangle$ pair is then propagated in the PTN, and the peers that receive this new log entry will either add it as a new entry or replace a previous entry by the same key.

C. Retrieving Usage Logs for Auditing

In 1997 the Inventor of the World Wide Web, Tim Berners-Lee, envisioned a browser button called 'Oh Yeah?' which is used to provide reasons why the user should trust the data [11]. We envisioned such a button on PETS, where users can ask 'Oh Who/What/Why/When/Where?' to determine the fate of their sensitive data. We implemented this functionality with the 'Audit' Button. An example of this button from the 'Transparent Health' system can be seen in Figure 5.

The sequence of actions that takes place when the 'Audit' button is clicked is represented in Figure 4. First the data owner, i.e. the agent that can prove ownership to the sensitive data to the data provider, makes an audit request. If the data owner is not already authenticated with the data provider, she will be redirected to be authenticated via the Verification Agent. Once authenticated, the data provider issues a get request to the PTN. The authentication processor in the PTN will validate the authenticity of the request, and if validated, will send the provenance log record for the sensitive data that was requested. Since the identity of the data consumer is known, the data owner can request for clarifications for the usage of the sensitive data from the data consumer. The data consumer can either be another user in the system or a process on the server. The process of clarifying the data access, use and transfers will also be logged in the PTN.

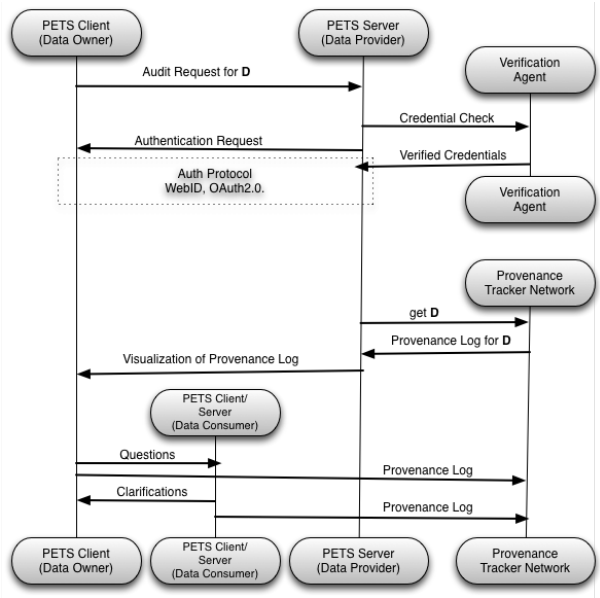

Fig. 4. Auditing Usage Logs with the PTN

D. Security Considerations and Abuse Prevention

In terms of security of the usage logs stored at individual nodes in the DHT, one can assume that there is a risk that a malicious node would be able infer the sensitive bits or even inject false information in the logs. However, the PTN requires that each usage log record to be encrypted by the agent that generated the log record. Therefore, even though the log record is stored in the malicious node, there is no guarantee that an attacker or the malicious node can read it and/or infer the sensitive data from the log records. Also, one could argue that the existence of logs may lead itself to privacy breaches. For instance, by observing the sequence of actions performed by a user, the adversary may be able to infer some sensitive information related to the activity of this user. Luckily, this is not possible with the PTN, as the keys for the usage logs are hashed by the URI of the sensitive data, and not by the user identifier. This reduces the likelihood of any single node obtaining the usage logs pertaining to a single user.

If a sensitive data item was used in a way that violates the usages restrictions set on them, then it is unlikely that the data consumers will report that usage as such to the data provider. However, all the actions within transparent systems are recorded in the provenance logs for the usage of sensitive data. Thus, by analyzing the logs, it can be inferred that even though the stated purpose of data usage is seemingly innocuous, whether the data consumer is in fact misusing the data or not. Similarly, abusive use of the protocol is possible if someone decides to a send massive number of requests for explanations through the 'Audit' functionality, resulting in denial of service attacks on the PTN. These kinds of attacks have plagued the track-back and ping-back systems in the past. However, unlike in those systems, the identity of the agents in the transparent system are known. Therefore, if someone issues too many requests, that person maybe banned from the PETS.

E. Decentralized vs Centralized Logging

The design of our system relies on decentralized logging of usage of sensitive data items. The choice of using a DHT for the PTN implementation was driven primarily by the fact that DHTs are inherently scalable, where any node can join and leave the network at any time. DHTs are also decentralized by design, thus ensuring the ownership of provenance log records are not controlled by a single entity and by design it is fault tolerant. A centralized design may very well work for reliable storage of usage data and to generate audit logs to enable privacy in a transparent manner. However, we argue that a decentralized design is better due to couple of reasons: (1) A node that participates in the PTN may also host a PETS application. But the usage logs stored in the Log Store may not be logs pertaining to the sensitive data from its own application. Any attempt to tamper the log records will result in a modified checksum, and the node is susceptible to be kicked out from the PTN for doing so. If the data is stored centrally, the log records can be changed at the hosting servers' discretion. (2) A decentralized architecture of the PTN replicate the data at many nodes, thus a failure of a single node would not affect the overall performance.

V. Motivating Use Case for PETS

Electronic Health Record (EHR) systems on the Web promise a wide variety of benefits and capabilities for healthcare. Health care providers can easily send and receive patient data necessary for treatment and analysis, and the patients themselves can use their data to track their health conditions through EHR systems. But the technologies that make these capabilities possible brings with them some undesirable drawbacks. Solutions through preventive measures often conflict with information requirements of care providers. Therefore, it is important to achieve a proper balance between these requirements to make health data accessible to patients. Furthermore, there is a plethora of free apps for nearly every health problem. Unfortunately, in the US for example, these apps are not covered by the privacy provisions of the Health Insurance Portability and Accountability Act or HIPAA , because they do not fall under the category of 'covered entities' as defined in HIPAA, unlike the health information shared directly between the patient and the doctor. Also, according to the HIPAA privacy rule, patients have the right to inspect and obtain a copy of their entire medical record with the exception of psychotherapy notes. A patient also has the right to an accounting of disclosures of protected health information made over the past six years [12]. However, the support for providing the data in an electronic medium is not that prevalent [13].

The 'agents' in our use case include, say, *Patient Peter, Doctor Dee, Steven Special, Pharmacist Precilla, and Insurance-agent Ira*. In addition to these human agents, *Patient Peter* also interacts with the free health app *MyHealth* that has *Peter*'s health information such as age, height, weight, blood pressure, cholesterol levels, vaccination data, medical conditions and medications both past and present. These agents operate in different systems. For example Doctor Dee who works at the General Hospital is Peter's primary care provider and Peter's primary health records are kept in a database at the General Hospital. Steven Special works at the Star Hospital, and Peter was recently referred to Steven Special

by Doctor Dee. Steven Special had to request all of Peter's medical records from General Hospital to get a comprehensive overview of Peter's conditions. However, the medical records pertaining to the referral visit was stored at a database in the Star Hospital. Pharmacist Prescilla, when filling the prescription, always looks at Peter's allergy information and past medications available from Peter's health record from the General Hospital, and now she refers to the records available from Star hospital as well. Insurance agent Ira receives health insurance claims from the General Hospital and Star Hospital for the procedures, as well as laboratory tests performed on Peter, and another claim from the Pharmacist for the medications. Depending on the health insurance policy, Ira may even request Peter's complete health profile to process the claim information. The MyHealth app might aggregate Peter's health information, daily activities, eating habits and sell all that to a third party.

As illustrated by the scenario above, there are complex information flows between various agents in these systems. There is always room for information misuse even if these agents are authorized to access, use and transfer the information by the data subject, i.e. Peter. Therefore, Peter might have a legitimate concern to ascertain that none of the agents in these systems use the information other than for the intended purpose, i.e. treatment. This concern might be aggravated if Peter is a celebrity and tabloids have an interest on his sensitive health information. The same applies if he is employed at the hospital, where his bosses and co-workers who have legitimate access to the system can pry on his private health information. Peter cannot enforce any preventive measures on the data usages as there could be emergency override situations where Peter might not be conscious or available to give meaningful consent for usage of his sensitive health data. However, if all accesses, usages and transfers of the data are recorded and are accessible to Peter, he will have a better trust in the system. Peter can use compliance checks after the fact to see if the agents who use his sensitive information have not violated any usage restrictions, or to see if there has been any mistakes. With such a transparency mechanisms in place, we can ensure that web based information management systems can be privacy preserving without being overly preventive.

VI. Evaluation

A. Transparent Health

To evaluate the viability of PETS we designed and implemented an electronic health records system called *Transparent Health*[2]. In this system, each access, use, and transfer on a health care record data marked as 'sensitive' is logged using the PTN implementation described in Sections III and IV. This system also models data from different systems. For instance, the patient's demographics and medical conditions data are stored at his primary care provider's information system, the medications information is stored at the pharmacist's information system, and the referral information is stored at the specialist doctor's information system. When a user joins Transparent Health, they can pull in data from these different systems to obtain a unified view as shown in Figure 5. Next to each 'sensitive' information there is an 'Audit' button that the user can obtain more information about that data usage. For example as can be seen in Figure 5 Peter Patient's medical report indicates that he has HIV/AIDS, a sensitive medical condition, and he might be interested in knowing if his medical conditions was only used in connection with his treatment purposes, and not for other purposes such as employment or insurance purposes.

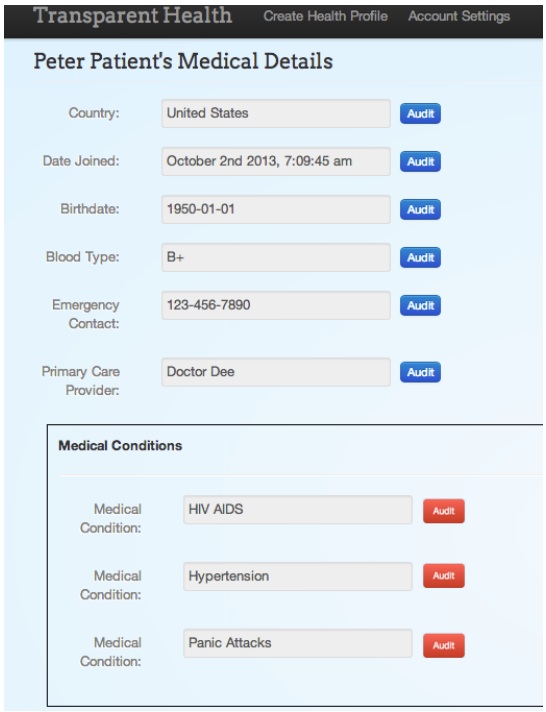

Fig. 5. Transparent Health: The complete health record of patients with the red 'Audit' button next to potentially sensitive information.

By clicking on the "Audit" button next to the HIV/AIDS medical condition, Peter can check to see how that has been consumed by the various agents in the system. Figure 6 shows an example audit log that will be displayed to Peter. It gives the date the event happened, who is responsible for the event, the role of that agent, and the purpose/intention as stated by the agent. If the patient thinks that this is a suspicious/unwanted access of their data, they can request for clarification from the agent by submitting a "Question". Although in the current system the compliance check is manual, we can imagine a system where explanations are generated for checking obligations fulfillment automatically as the logs are generated.

B. User Study on Transparent Health

One of the primary enablers of PETS is the ability to determine when, where, how, what, why an individual's privacy was violated, and who is responsible for the violation. Therefore, by using Transparent Health as a test bed, we conducted a qualitative study on user perceptions about having access to usages of their data in such a transparent manner.

1) Participant Profiles and Preliminary Questions: We recruited 25 participants for the study (17 male and 8 female, ages ranging from 18-55). All of these participants indicated that they visit a health care provider at least once a year with

[2]Transparent Health is available at *http://www.transparent-health.us* for demonstration purposes.

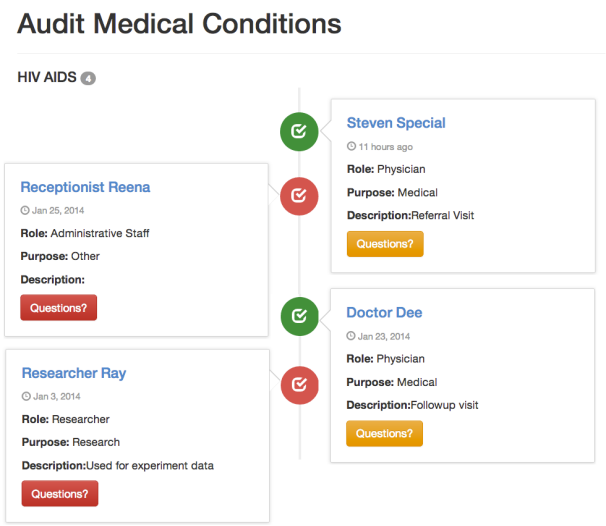

Fig. 6. Audit Logs indicating how a sensitive data was used in Transparent Health. Depending on the usage restrictions set by the user, all the questionable usages appear on the left. The user can submit a question about the respective usage and subsequently flag the usage as a privacy violation.

the median number of visits being 5. 20 participants indicated that they have access to their health care records after a visit to their doctor through an online health care portal. From the participants who indicated that they do not have access to their health data after a visit to the doctor, all but one expressed interest in using a system as such. Then we asked them if they are worried about their sensitive health information being misused in electronic health care record systems. 15 answered yes, 8 answered no, and 2 answered that they do not care. Then afterwards, we gave some background as to how their private health data can be misused. Provided that there are means to figure out a privacy violation, we asked them what they consider most important in knowing: (1) Who? : the identity of the personnel that misused their information, (2) When? : the time at which the information misuse happened, (3) How? : how did they have access to the information and how they misused the information, (4) Where? : from where did they get access to the information, and where did they send the information to, (5) Why? : the motivations behind the data misuse, and (6) What did they misuse?. We specifically asked them to categorize their responses as **Rank 1**, **Rank 2** and **Rank 3**. The results are summarized in Figure 7. The results suggest that most users are interested in knowing 'who' misused the information, followed by 'how' the misuse happened, and 'where' the violators got access to the sensitive health information. This validated our implementation design decision of requiring agents in PETS to have a unified identity so that they could be identified in case of a violation. Also, the provenance trail is designed to provide enough evidence of other conditions in a misuse.

2) Creating the Health Profile: After interviewing the users, we asked them to try out our transparent personal health information system available at http://www.transparent-health.us. The first task they had to do was to create their health profile in the system without revealing their sensitive personal health data (nothing prevented them from entering their personal data, but we advised them not to add anything too

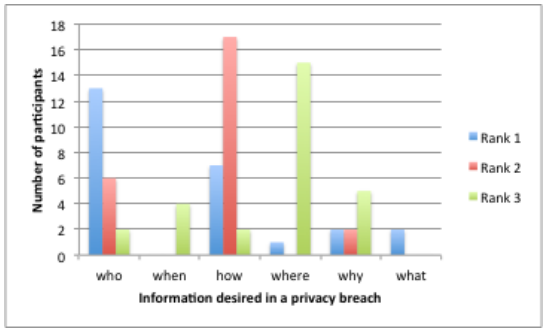

Fig. 7. Categorization of what users consider most useful in knowing if there is a mechanism to figure out privacy breaches of their sensitive information

personally revealing). We presumed that self created health profiles would give the participants a better sense of privacy awareness, rather than giving them canned health profiles that they are not able to identify with. To help the users with the process of creating the profile, we provided them with a health information profile creation guide. The guide suggested several medical conditions that they can choose from to add to the past and current medical conditions. The guide also provided the ability to add any other illnesses the users want in their profile. Based on the illnesses that were selected before, a medication list was provided. They had the option of removing some of the medications and adding some other medications. We also asked them the names of their primary care providers, and a specialist that they were referred to, to provide more identifying context for the user.

3) Setting Usage Restrictions: For each of the data item the participants entered, they had the option to mark that as a sensitive data item. An example would be to mark a health condition such as 'HIV AIDS' as sensitive. To do this they were presented with an interface to select (1) with whom they would like to share this information with (i.e. researchers, insurance companies, affiliates, and non-medical staff such as hospital receptionists, etc.) and (2) for what purposes (i.e. research, insurance claim processing, marketing, or other purposes). Please note that the physician, nurse and pharmacist roles in Transparent Health all have comprehensive access to the patient's health record by default, but they have to specify the purpose when accessing any data marked as sensitive.

4) Simulating Information Flows: Based on the information provided, we simulated several scenarios asking the users to acknowledge that the events in the scenarios happened. Examples of these events include: the doctor diagnosing one of the illnesses the user had picked, the user picking up the medications from the pharmacist, the doctor referring the user to a specialist, the participant agreeing to contribute the personal medical data for a research experiment by signing a waiver, etc. These events simulated some of the real world events that may have happened with the user knowing about them. As these events were being acknowledged by the user, the corresponding usage logs were generated and added to the PTN. We also added two other random events, the first event can be construed as a misuse of the patient's private health information such as transfer of the medication information to a marketing firm by the pharmacist, and the other event is a

treatment related activity that the user was not aware of such as referral event where the doctor is sending the medical record to a specialist.

5) Auditing their Health Records: After the users finished completing their health profile, we asked them to test out the functionality of the 'Audit' button. Their task was to go to their health profile, and select the data fields they marked as sensitive and flag the privacy violations as identified by PETS are indeed privacy violations. An example audit log is given in Figure 6. PETS does a trivial inference based on the usage restrictions set on the sensitive data items by the participants to identify potential privacy violations. One of the random events we added, as described in the previous section, was designed to be a misuse of the data potentially leading to a privacy violation. 21 of the participants indicated that they like the feature of being able to see how their information was used by those who are authorized to work with their personal health information. When asked if they feel that the synthetic privacy violation events were indeed privacy violations from their perspectives, 18 said yes, 3 said no, and the other participants said they do not mind if those agents viewed or used their data in that potentially privacy intrusive way.

6) Reversing the Roles: Next, we asked if they would agree to use a system such as Transparent Health if they were to take the roles of health workers that can be audited by the patients. 16 participants said yes, whereas the rest said that they would only use such a system, only if it was mandated by a law. Many participants indicated that even though as a health worker they would have to be more conscientious of their actions, the patient has a right to the information.

7) Our Hypothesis and Supporting Anecdotes: It was our hypothesis that users will have a better understanding about their overall health care, and have a better confidence in electronic health care systems since they will be able to see if there has been any unwarranted accesses and usages of their protected private health information. Here are some of the anecdotes from the participants from the user study that supports our hypothesis: "A very innovative thought! This kind of site will be indispensable after few years.", "Auditing my health information is easy from Transparent Health", "It is a good system to ask questions from the doctors about my health information".

VII. Related Work

The PTN uses the W3C provenance ontology recommendation [10] in defining the terms used in the information flow of PETS. There has been decades of research on enabling provenance in scientific workflows, but up until recently, provenance concepts have not been applied in relation to preserving privacy in systems. With the arrival of the provenance ontology various tools have emerged that are designed to preserve the provenance of data in software systems. Provenance management tools such as 'ProvToolBox' [14] creates Java representations of the prov data model and enables manipulation of it from the Java programming language. ProvenanceJS [15] is a Javascript tool that can embed and extract provenance from HTML pages. Although these tools generate provenance, they have not been applied in the context of tracing breaches of information privacy like the PTN is designed to do in PETS.

Structured logging, i.e. generating log files that incorporate dependencies between entities, the start and stop times of activities, and the inputs/outputs of activities, has been a topic of interest in many programming languages research [16]. There are many infrastructures, specifically in web service adaptors, that can be repurposed for collecting provenance [17]. Samavi et al describe a framework designed to facilitate privacy auditing through the use of two ontologies, whereby one provides provenance enabled logging of events, and the other for synthesizing the contextual integrity for compliance and observation derivation [18]. In comparison to our architecture, these systems have not implemented an end-to-end infrastructure such as the PTN in tracking privacy violations.

In relation to privacy in health care systems, there have been some work on providing transparent and accountable data access in health organizations [19]. Their approach is to give unrestricted access to legitimate users and channel usage inquiries and usage justifications through an information accountability service. Unlike in our system where the focus is on enabling a decentralized architecture to find breaches across systems and notify them to the data owner, their focus is on implementing a centralized architecture for individual health care providers. The patients in their systems can seek redress within the system and penalize misbehaving actors by restricting access to the records.

The need for transparency tools is motivated in [20] and [21]. These two papers have surveyed how the appropriate use of data is monitored in several technical projects such as Privacy Bird [22], and Privacy Evidence [23]. Our work builds on some of these work, especially the TAMI project [24] where the focus is to log the usages to determine the appropriate usage of sensitive data.

VIII. Future Work

We have presented a technical solution to a social problem in this paper. One of the main challenges we foresee is the reluctance for individuals to behave freely knowing that accesses and usages of their records are logged continuously. Thus, in a future iteration of this project, we hope to borrow the notion of "accountable anonymity" as used in [25] where a participant remains anonymous unless he breaks the rules and disrupts the system, at which point his identity is revealed.

Even though we do not yet have a use case that implements the PTN in a production setting, we have done a scalability test on a PTN comprising of 100 nodes on PlanetLab. We continuously assessed the latency of requests to update and retrieve usage logs based on a synthetic workload over the course of 24 hours. As the size of the log grows, the time remained almost comparable for our workload. However, we have not yet done a stress test of the PTN under very large number of requests on very large scale of usage log data. For future work, we are working on a graph summarization algorithm that can suppress triples that were generated before a certain epoch and create a summary, while leaving the newer additions in triple format to answer any audit request.

There are many healthcare systems out there and there is no apparent incentive for healthcare providers to adopt the PETS architecture. However, at least in the US, there are slow but steady push towards opening up personal health records

to patients via the 'Blue Button Initiative'. The data from blue-button enabled sites are meant to increase interaction among healthcare providers and other trusted entities. Since the data will no longer locked up in silos, we can imagine a decentralized healthcare data eco-system evolving that entails complex usage scenarios. In such a system, which may come to fruition in the near future, PETS can be readily adopted.

IX. Conclusion

Privacy without proper security is impossible. However, the Web provides a very easy medium to access, copy and transfer sensitive information through services at users fingertips both for legitimate purposes and malicious purposes resulting in many data breaches and violations of privacy even though there are rigid security controls. Therefore, there is a need for safeguards that supplement traditional access control mechanisms, especially in situations where access control will be overly prohibitive in providing access to data in crucial decision making processes. Therefore, in this paper we have presented PETS that focusses on transparency of access and usage activity to patients in an EHR system while providing good security measures with robust authentication mechanisms and strong encryption. PETS makes transparency a first class citizen in information systems. This enables the data owner to check how her data has been used. The usage data can be reasoned with individual, organizational, state or federal policies or usage restrictions to assert that no violation has taken place. Therefore the data subjects will have more trust in the PETS while the data consumers will act appropriately and be less likely to misuse data.

This paper stresses on implementing transparency to achieve accountability with provenance mechanisms. The logging process in PETS adds accountability to the system in order to enhance privacy by offering the possibility to check a posteriori that a privacy leak has occurred. We put forward a proposal for the underlying architecture for PETS using a global network of peer servers dedicated to preserving provenance of data and usages called the Provenance Tracking Network, along with an implementation for PETS in the healthcare domain called Transparent Health. We evaluated how effective this architecture is, in enabling patient privacy with a user study on Transparent Health. The results from the evaluations demonstrated that this architecture is promising, and there is lot of interest for transparent web applications from users who are interested in protecting their privacy from unintended leakages of their sensitive data.

X. Acknowledgements

This material is based on work supported by the National Science Foundation under Grant Number 1228687.

References

[1] J. Staddon, P. Golle, and B. Zimny, "Web-based inference detection," *SS*, vol. 7, pp. 1–16, 2007.

[2] L. Kagal and H. Abelson, "Access control is an inadequate framework for privacy protection," in *W3C Privacy Workshop*, 2010.

[3] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G. J. Sussman, "Information accountability," *Commun. ACM*, vol. 51, no. 6, pp. 82–87, Jun. 2008. [Online]. Available: http://doi.acm.org/10.1145/1349026.1349043

[4] J. Feigenbaum, J. A. Hendler, A. D. Jaggard, D. J. Weitzner, and R. N. Wright, "Accountability and deterrence in online life," in *Proceedings of the 3rd International Conference on Web Science, ACM*, 2011.

[5] O. W. Seneviratne, "Augmenting the web with accountability," in *Proceedings of the 21st international conference companion on World Wide Web*. ACM, 2012, pp. 185–190.

[6] O. Senevitane and L. Kagal, "Addressing data reuse issues at the protocol level," in *Policies for Distributed Systems and Networks (POLICY), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 141–144.

[7] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, "Opendht: a public dht service and its uses," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 73–84.

[8] S. Tramp, H. Story, A. Sambra, P. Frischmuth, M. Martin, S. Auer *et al.*, "Extending the webid protocol with access delegation," in *Proceedings of the Third International Workshop on Consuming Linked Data (COLD2012)*, 2012.

[9] D. Hardt, "The oauth 2.0 authorization framework," 2012.

[10] T. Lebo, S. Sahoo, and D. McGuinness. (2003) Prov-o: The prov ontology. [Online]. Available: http://www.w3.org/TR/prov-o/

[11] T. Berners-Lee. (1997) The "oh, yeah?" -button. [Online]. Available: http://www.w3.org/DesignIssues/UI.html#OhYeah

[12] G. J. Annas, "Hipaa regulations-a new era of medical-record privacy?" *New England Journal of Medicine*, vol. 348, no. 15, pp. 1486–1490, 2003.

[13] A. K. Jha, T. G. Ferris, K. Donelan, C. DesRoches, A. Shields, S. Rosenbaum, and D. Blumenthal, "How common are electronic health records in the united states? a summary of the evidence," *Health Affairs*, vol. 25, no. 6, pp. w496–w507, 2006.

[14] S. Magliacane, "Reconstructing provenance," in *The Semantic Web–ISWC 2012*. Springer, 2012, pp. 399–406.

[15] P. Groth, "Provenancejs: Revealing the provenance of web pages," in *Provenance and Annotation of Data and Processes*. Springer, 2010, pp. 283–285.

[16] B. Tierney, W. Johnston, B. Crowley, G. Hoo, C. Brooks, and D. Gunter, "The netlogger methodology for high performance distributed systems performance analysis," in *High Performance Distributed Computing, 1998. Proceedings. The Seventh International Symposium on*. IEEE, 1998, pp. 260–267.

[17] P. Groth, S. Miles, and L. Moreau, "Preserv: Provenance recording for services," 2005.

[18] R. Samavi and M. P. Consens, "L2tap+ scip: An audit-based privacy framework leveraging linked data," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*. IEEE, 2012, pp. 719–726.

[19] R. Gajanayake, R. Iannella, W. B. Lane, and T. R. Sahama, "Accountable-ehealth systems : the next step forward for privacy," in *1st Australian eHealth Informatics and Security Conference*, November 2012. [Online]. Available: http://eprints.qut.edu.au/55217/

[20] H. Hedbom, "A survey on transparency tools for enhancing privacy," in *The future of identity in the information society*. Springer, 2009, pp. 67–82.

[21] M. Janic, J. P. Wijbenga, and T. Veugen, "Transparency enhancing tools (tets): an overview," in *Socio-Technical Aspects in Security and Trust (STAST), 2013 Third Workshop on*. IEEE, 2013, pp. 18–25.

[22] L. F. Cranor, P. Guduru, and M. Arjula, "User interfaces for privacy agents," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, no. 2, pp. 135–178, 2006.

[23] S. Sackmann, J. Strüker, and R. Accorsi, "Personalization in privacy-aware highly dynamic systems," *Communications of the ACM*, vol. 49, no. 9, pp. 32–38, 2006.

[24] D. J. Weitzner, H. Abelson, T. Berners-Lee, C. Hanson, J. Hendler, L. Kagal, D. L. McGuinness, G. J. Sussman, and K. K. Waterman, "Transparent accountable data mining: New strategies for privacy protection," 2006.

[25] H. Corrigan-Gibbs and B. Ford, "Dissent: accountable anonymous group messaging," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 340–350.