Publishing L2TAP Logs to Facilitate Transparency and Accountability

Reza Samavi
University of Toronto
samavi@mie.utoronto.ca

Mariano P. Consens
University of Toronto
consens@mie.utoronto.ca

ABSTRACT

We propose publishing L2TAP privacy logs to facilitate privacy auditing tasks that involve multiple auditors, an increasingly common requirement in the context of social computing and big data driven science. Our proposal utilizes two ontologies, L2TAP and SCIP, designed for deployment in a Linked Data environment. L2TAP provides provenance enabled logging of events. SCIP synthesizes contextual integrity concepts to express key privacy-related semantics associated with log events. We describe SPARQL query-based solutions for privacy log construction, obligation derivation, and compliance checking. The solutions facilitate accountability and transparency among participants (privacy auditors in particular).

1. INTRODUCTION

The protection of individuals' privacy is becoming increasingly more challenging in the era of social computing and data driven science. While privacy protection has implications in many application areas, it is clearly challenging when health related data is involved. Big data enabled biological and biomedical research involves massive datasets of human genome, biological imaging, and clinical information collected and aggregated from individual health records. Protecting data subjects' privacy in clinical research is a concern addressed by multiple legislations and regulations. For example, the U.S. Department of Health and Human Services (HHS) [27] obliges investigators to protect the privacy of data subjects and to maintain the confidentiality of data. HHS also requires investigators to establish oversight mechanisms and monitoring plans for research projects involving human subjects, and to remain accountable to the subjects' privacy rights. Auditing is essential to the enforcement of accountability, and many scenarios involve auditors from multiple institutions that monitor the fulfillment of privacy obligations.

To illustrate the need for an audit mechanism that facilitates accountability and transparency among multiple participants, consider a research study that analyzes the primary reasons for intensive care unit (ICU) hospitalization, examining the effectiveness of different types of medications across patient demographics (sex, age, and ethnicity). The scenario is depicted in Fig. 1. The dataset used for the study is MIMIC II, a Clinical Database provided by PhysioNet [10], where records contain information about the ICU admission of patients [16]. Although MIMIC II database is a de-identified public dataset, the access is available only under terms of a data use agreement (DUA[1]). This DUA defines a list of obligations that a researcher agrees to fulfill. Some of these obligations involves purposes and roles. For example, *the dataset should be used only for academic research purposes and by a researcher* (ob_1). Some others are pre-obligations which are actions that need to be performed prior to access. For instance, the DUA states *a researcher should complete a training program in human research subjects protections prior to access* (ob_2). There are also some post-obligations which are actions that need to be performed after access has been granted such as: *If the researcher finds information within restricted data that she believes might permit identification of any individual, she will report the location of this information promptly by email* (ob_3).

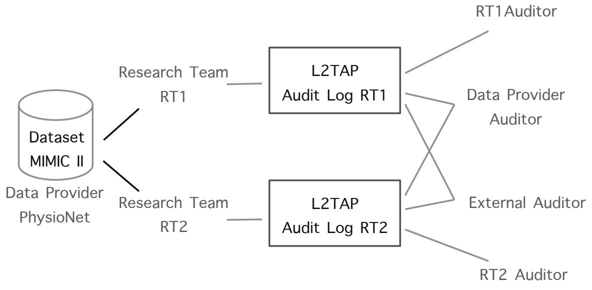

Figure 1: Privacy Auditing Scenario

Two research teams (RT1 and RT2) are collaborating on this study. The teams could be based on related or unrelated research institutions (i.e., RT1 is based on a hospital, while RT2 is based on a university department, and the hospital could be part of the university, or not). The privacy policies mentioned above govern the access to MIMICII dataset. The policies are designed by PhysioNet according to HIPPA [26] and other privacy regulations in order to protect privacy

[1] http://physionet.org/works/mimic2cdb/access.shtml

of individuals whose data are used in research studies. In order to check if the research teams are compliant to these policies multiple potential auditors must be able to audit the log of access and fulfilment of obligations. One of the potential auditors is the Data Provider itself who oversees the contract to ensure the usage of data is accordance to the agreement. The auditors from two teams may also want to audit the process with respect to the internal data protection policies. In addition, an external auditor should be able to ensure the research involving individuals health data are fully HIPPA compliant.The challenge needs to be addressed is that while multiple participants are involved in generating privacy logs (e.g. the data provider, the research teams, and the researchers), all potential auditora should be able to check the log to see if the researchers are respecting privacy of data subjects.

In the past few years, we have observed multiple practical proposals with focus on privacy of big datasets and linked data ([22, 18, 7, 8]). The goal of these studies have been on access control frameworks that define who can access which resources. They achieve privacy through safeguarding of data before the access is granted and provide no solutions for privacy support after the access.There are also solid theoretical and practical work to support privacy auditing (data usage control after access is granted). However they either exploit complex logic (e.g. [2, 9, 3, 5]) that jeopardizes their practical benefits or use system level logging standards (e.g.[15, 6]) to generate privacy audit logs on an application by application basis, thus generating privacy logs not exploitable by multiple participants and auditors in an heterogenous environment.

In [23] we proposed L2TAP ontology (Linked Data Log to Transparency, Accountability and Privacy) that allows participants to log in RDF [28] the provenance assertions of privacy related events. We also proposed a second pluggable ontology SCIP (Simple Contextual Integrity Privacy) to capture the privacy semantics of log events and enable SPARQL query-based implementations of auditing and compliance checking in a personalized health workflow[2].The scalability of the framework for compliance checking has been evaluated by a set of queries described in [23].

Using L2TAP+SCIP, this paper proposes a standard way of privacy auditing in the big data research context. We propose multiple SPARQL query-based solutions (with a limited reasoning support of RDFS) to facilitate the tasks of constructing L2TAP privacy logs (when privacy policies are applicable to the classes of individuals and data items), deriving obligations from privacy policies, and compliance checking. If research teams agree on the semantics of publishing the log based on L2TAP+SCIP, they can show to the auditors their compliance in only one effort and auditors can oversee the compliance of parties involved without additional efforts. In other words, after the log has been created and all obligations and their fulfillments are captured the research team can check the log and provide it as an evidence of accountability [21]. Using the same log and the SPARQL solutions, all other auditors including the data provider's auditor, the institution auditors (described in [26] as Institutional Review Board (IRB) for a single-site research and Data and Safety Monitoring Board (DSMB) for multi-site research), and the external auditor can check compliance.

The paper structure and contributions are as follows. Section 2 provides an overview of L2TAP and SCIP and shows how the ontology can be used to capture the log events and their privacy semantics. Section 3 describes our SPARQL query-based solutions for constructing the log, obligation derivation, and compliance checking. Section 4 describes the related research. We conclude in Section 5.

2. L2TAP LINKED DATA LOG

In this section we first motivate the need for two ontologies to generate privacy audit logs. Then using our motivating scenario we describe L2TAP, an ontology for specifications of the header of privacy log events, and SCIP, an ontology that provides necessary specifications to encode privacy semantics of the body of log events.

The goal of L2TAP is to provide a set of classes and properties that can be used to represent and publish a log of *privacy events* as Linked Data. In the motivating scenario expressing access policies and obligations, requesting access to the dataset, fulfilling obligations are some of the typical privacy events that we expect L2TAP to be able to capture.

L2TAP follows the principles of Linked Data [11] to publish logs. Everything in the `l2tap:Log` is expressed in terms of some `l2tap:LogEvent`s. URIs are used as names for logs, log events, participants, and processes. Thus, the log events can be published as web dereferenceable URIs by participants. Participants who want to dereference a published log are authenticated and communicated via a secure `https` channel. After we describe L2TAP and SCIP ontologies, at the end of Subsection 2.2 we will provide justification on why these ontologies rely on dereferenceable URIs and how Linked Data infrastructure allows to achieve log data integration when multiple parties contribute into the log their privacy events in different points in time.

The L2TAP ontology describes the header of a log event and is scoped to answer the provenance queries about log events, such as *who* has contributed an event to the log and *when*. The *when* in L2TAP can be expressed as simple `xsd` time using two L2TAP properties, `l2tap:eventTimestamp` and `l2tap:publishingTimestamp`. There are some subtlety in capturing the *who* in L2TAP. Following the second principle of Linked Data, using `http://` URIs as names for participants, amounts to a data publisher choosing part of an `http://` namespace that the publisher controls, by virtue of owning the domain name [11]. In L2TAP, the publisher of the log events is the logger who owns the domain of the log and can talk about the events and their assertions (e.g. `https://logRT.org`). If an L2TAP logger wishes to identify a participant as the *who* in an event header, the logger must register the participant, i.e. mint the URI of the participant with the namespace in its domain. Registered participants will be considered accountable for the assertions that they make in the log.

The privacy semantics of privacy events (e.g. what is an obligation fulfilment) are contained in the body of a log event and expressed using the SCIP ontology. In designing SCIP,

[2] L2TAP and SCIP are documented at `http://l2tap.org`.

```
 1 <https://logRT.org> a l2tap:Log.
 2 <https://logRT.org/logevent/e1> a l2tap:LogInitializationEvent;
 3   l2tap:initializesLog <https://logRT.org>;
 4   l2tap:logger <https://RT.org/logger>;
 5   l2tap:publicationTimestamp "2014-01-26T12:00:00Z"^^xsd:dateTime;
 6   l2tap:timeline <https://RT.org/sitetime>.
 7 <https://RT.org/logger> a foaf:Agent.
 8 <https://RT.org/sitetime> a l2tap:Timeline;
 9   l2tap:physicalTimeline tl:universaltimeline;
10   l2tap:clock "wwp.greenwichmeantime.com/" ^^ xsd:string;
11   l2tap:clockSyncFreq [tl:duration "P7DT"^^ xsd:duration].
```

Figure 2: Log initialization event

we are inspired by the contextual integrity (CI) perspective [20]. The SCIP ontology provides mapping targets for basic notions of participants in an information flow, privacy contexts, and privacy norms as described in CI. The goal of the SCIP ontology is to define a minimum set of classes, properties and constraints that allows the basic compliance queries (e.g. which access request is non-compliant?) to be answered using SPARQL queries.

Having two namespaces is the basis for the framework flexibility and extensibility. The proposed SCIP ontology is just one instance of a class of pluggable ontologies to express privacy semantics and can be substituted with an ontology with more-or-less expressive power without impacting the semantics of the log header.

2.1 Log Event Types

L2TAP specifies three types of log events, *log initialization events*, *participant registration events*, and *privacy events*.

Log Initialization Events. This type of events defines which `l2tap:Log` is being initialized using the `l2tap:initializesLog` property. It also records assertions on the log characteristics such as who the logger is (`l2tap:logger`), and how the event timestamps are captured (`l2tap:logClock`). Fig. 2 provides an example of using the L2TAP ontology to encode a log event that initializes a log with `https://logRT.org` URI (line 1). This privacy log has a logger with `https://RT.org/logger` URI (line 4). The logger is a `foaf:Agent`[3] (line 7). The physical timeline for this log is a constant in the timeline ontology[4] (line9). Lines 10 and 11 encode the log's reference clock and the syncing frequency.

Participant Registration Events. This event type is used to register a `foaf:Agent` as an L2TAP log participant who can then submit the future log events. The `l2tap:registersAgent` property links a log event to a `foaf:Agent` who will be recognized by the logger as the registered agent. As described above the participant registration event marks the time instant that a participant's URI has been minted in the logger's domain. So the registered participants will be kept accountable with respect to the log events that they are contributing to the log in the future. In our scenario research teams as receivers of data and PhysioNet as the data provider are participants in the log. If the data was not anonymized each individual data subject or a class of data subjects could have also been registered as participants.

```
 1 https://logRT.org/logevent/e2> a l2tap:ParticipantRegistrationEvent;
 2   l2tap:memebrOf <https://logRT.org>;
 3   l2tap:eventTimestamp "2014-01-27T12:00:00Z"^^xsd:dateTime;
 4   l2tap:publicationTimestamp "2014-01-27T12:00:01Z"^^xsd:dateTime;
 5   l2tap:registersAgent <https://RT.org/ClinicalResearchers>;
 6   l2tap:eventParticipant <https://logRT.org/participants/RT>.
 7   l2tap:eventData <https://logRT.org/logng/ng2>.
 8 <https://logRT.org/participants/RT> a l2tap:Participant;
 9   l2tap:registeredAgent <https://RT.org/ClinicalResearchers>.
10 <https://RT.org/ClinicalResearchers> a foaf:Agent.
11 <https://logRT.org/logng/ng2>={
12 <https://RT.org/ClinicalResearchers/RT1> a <https://RT.org/
        ClinicalResearchers> .
13 <https://RT.org/ClinicalResearchers/RT2> a <https://RT.org/
        ClinicalResearchers> .
14 <https://RT.org/ClinicalResearchers/Mark> a <https://RT.org/RT1> .
15 <https://RT.org/ClinicalResearchers/Joe> a <https://RT.org/RT1> .}
```

Figure 3: Participant registration event

Fig. 3 shows an example of using the L2TAP ontology to register the class of research teams as a `foaf:Agent` with `<https://RT.org/ClinicalResearchers>` URI (line 5). Note that this is the URI of a class of researchers. In lines 8-9 the `l2tap:Participant` class and the `l2tap:registeredAgent` property are used to capture the fact that the URI of the class of researchers is minted in the logger's domain. It is optional for a participant registration event to use the `l2tap:participantData` property and add a named graph [4] as the event data (payload) to the event. Suppose in our motivating scenario RT1 and RT2 are two classes of researchers. RT1 uses the dataset to study patients under 18 and RT2 studies patients 18 year and older. The optional named graph can be used to capture this classifications and additional information about the members of each class. For example we used the named graph (lines 11-15) to encode research team hierarchy and memberships. Therefore the accountability can be cascaded to a specific individual.

Privacy Events. A *privacy event* is used to encode privacy processes such as expressing privacy policies, access requests, and obligation fulfilment. Fig. 4 shows how the L2TAP ontology is used to log provenance assertions of privacy policies applicable to our scenario. The quads in this privacy event are grouped in two sets. The quads in lines 1-6 are the header of the event and the quads in line 8 onward are the body of the event. The data provider (PhysioNet) is the one *who* submits the quads of the policies to the log (line 3). The body of this log event (wrapped in `https://logRT.org/logng/ng1` named graph) describes privacy policies and preferences as the payload of the event. The SCIP ontology is used to express the semantics of a privacy event's body.

2.2 Log Event Privacy Semantics

The L2TAP ontology described so far encodes a privacy event and its accountable participant regardless of the privacy semantics of the event. The SCIP ontology provides necessary vocabularies to capture the privacy semantics. We categorize the semantics in four groups: privacy preferences (policies), access requests and responses, obligation fulfillments, and access activities.

Privacy Preferences. The `scip:PrivacyPreference` class is used to encode a context and the norms applicable to the context. The context in SCIP is characterized using multiple

[3] http://xmlns.com/foaf/spec/

[4] http://purl.org/NET/c4dm/timeline.owl#

```
1  <https://logRT.org/logevents/e3> a l2tap:PrivacyEvent;
2    l2tap:memebrOf <https://logRT.org//>;
3    l2tap:eventParticipant <https://logRT.org/participants/PhysioNet>;
4    l2tap:eventTimestamp "2014-01-28T12:00:00Z"^^xsd:dateTime;
5    l2tap:publicationTimestamp "2014-01-28T12:01:00Z"^^xsd:dateTime;
6    l2tap:eventData <https://logRT.org/logng/ng1> .
7  <https://logRT.org/logng/ng1> = {
8  <https://logRT.org/PhN_pp1> a scip:PrivacyPreference; ...}
```

Figure 4: The header of a privacy event (for policies)

```
 9    scip:expressedBy <https://logRT.org/participants/PhysioNet>;
10    scip:hasValidity [time:hasBegining "2014-01-01T00:00:00Z";
11                      time:hasEnd "2015-01-01T00:00:00Z"];
12    scip:dataItem <https://mimicii.org/patients/MEDITEM>;
13    scip:requestorRole <https://RT.org/roles/scientific_researcher>;
14    scip:purpose <https://RT.org/purposes/scientific_research>;
15    scip:privacyPrivilege <https://RT.org/privileges/read>;
16    scip:obligation <https://RT.org/obs/ob2>;
17    scip:obligation <https://RT.org/obs/ob3>;
18    scip:propositionalExpression <https://RT.org/exp/phy1> .
19  <https://RT.org/obs/ob2> a scip:ObligationTemplate;
20    scip:performAction <http://ontology.org/actions/
          obtain_training_certificate>;
21    scip:occurrenceGap "-1"^^xsd:integer;
22    scip:performanceDuration "1"^^xsd:integer.}
```

Figure 5: The body of a privacy event (for policies)

classes: `scip:DataItem`, `scip:Purpose`, and `scip:PrivacyPrivilege`. Use, collect, and disclosure are different types of privacy privileges. Participants in a context interact with each other in certain capacities or roles. In SCIP, roles of three main participants in an information flow are encoded using `scip:dataSubjectRole`, `scip:dataRequestorRole`, and `scip:dataSender Role` properties. The `scip:Role` class is used to capture the abstract and concrete roles. In SCIP, roles, purposes, data items and privacy privileges are represented as lattice using `rdfs:subClassOf`.

Fig. 5 shows how the SCIP ontology is used to encode the obligations described in our scenario. Note that the quads in this figure are the continuation of the quads in Fig. 4. Line 9 describes by whom the privacy preferences are expressed using the `scip:expressedBy` property. Note that the minted PhysioNet URI is the participant who submits the privacy policies. The quads in line 10 and 11 describe the validity time interval of the policies. The first obligation in our scenario (ob_1) is expressed as legitimate purpose (line 14) for using the dataset (line 12) and the acceptable roles of participants (lines 13) and the privilege that will be granted if the obligations are fulfilled (line 15).

There are also norms associated with a context that describe obligations or actions that need to be performed before (pre-obligation) or after (post-obligation) the dataset is accessed [19]. The `scip:ObligationTemplate` is a subclass of `scip:Obligation` that captures these actions. Obligations, expressed in privacy preferences, are templates for future instantiation of executable obligations. The `scip:Obligation` is `rdfs:subClassOf scip:ObligationTemplate`. Obligations has properties to express temporal constraints associated with an obligation. For example, the second obligation requires taking the training course (`obtain_training_certificate`) prior to access. This obligation is encoded using `scip:performAction` (line 20). `scip:occurrenceGap` property in line 21 encodes the relative time interval for performing the obligation (a positive integer indicates occurrence *after* the access activity, and a negative integer *before*). The `scip:performanceDuration` property in line 22 encodes the time required to perform the obligation. The third obligation is a post-obligation and requires to be fulfilled after access has been granted and when a record deem to be identifiable.

```
 1  <https://logRT.org/logevents/e4> a l2tap:PrivacyEvent;
 2    l2tap:memebrOf <https://logRT.org>;
 3    l2tap:eventParticipant <https://RT.org/ClinicalResearchers/Mark>;
 4    l2tap:eventTimestamp "2014-01-29T12:00:00Z"^^xsd:dateTime;
 5    l2tap:publicationTimestamp "2014-01-29T12:01:00Z"^^xsd:dateTime;
 6    l2tap:eventData <https://logRT.org/logng/ng2>.
 7  <https://logRT.org/logng/ng2> = {
 8  <https://RT.org/requests/req1> a scip:AccessRequest;
 9    scip:dataRequestor <https://RT.org/ClinicalResearchers/Mark>;
10    scip:dataSender <https://RT.org/participants/physionet>;
11    scip:dataSubject <https://mimicii.org/patients>;
12    scip:dataItem <https://mimicii.org/MEDITEM>;
13    scip:purpose <https://RT.org/purposes/clinical_research>;
14    scip:requestorRole <https://RT.org/roles/clinical_researcher>;
15    scip:requestedPrivilege <https://RT.org/privileges/read> .}
```

Figure 6: Log event for access request

Access Requests and Responses. The `scip:AccessRequest` class is used to encode a request by a researcher to access a dataset. A number of classes that we used to express privacy policies (such as `scip:DataItem`, `scip:Role`, `scip:Purpose`, `scip:DataItem`, and `scip:DataRequestor`) will also be used to express access requests. An access request can be initiated by a class of participants or an individual participant. In our motivating scenario, we assume one of the researchers in the team (Mark) uses the framework to log its access request (cf. Fig. 6). Note that the *who* in the header of this log event is Mark's URI (line 3) who is a member of clinical researcher class. Line 8 encodes the Mark's access request as an instance of `scip:AccessRequest`. The `scip:dataRequestor` property in line 9 captures the URI of the data requestor (Mark), `scip:dataSender` (line 10) captures who should send the data (PhysioNet) while `scip:dataSubject` (line 11) captures whose data has been requested (Patients class in MIMIC II dataset). Similar to the privacy policies, we encode in line 12 the URI of requested data items (MEDITEMS: class of all medications taken by patients), the purpose for accessing data (line 13), and the roles of the participants requesting access (line 14). The privacy privilege that has been requested is encoded by `scip:requestedPrivilege` in line 15.

The `scip:AccessResponse` class encodes the boolean response to an access request as well as the applicable obligations. The log event shown in Fig. 7 records the access response to the Mark's request by dereferencing the corresponding access request URI (line 9). Line 10 encodes the access decision. Associated with each access response there could be a set of applicable obligations. The quads in lines 11-17 encode one of the obligations derived from privacy policies applicable to the study. Lines 11 in this listing refers to the URI of the corresponding obligations using `scip:contextObligation`. When multiple obligations arise from an access request, a propositional formula φ describes how the satisfaction of these obligations relates to the overall compliance of the access request. In our example scenario $\varphi \equiv ob_1 \wedge ob_2 \wedge ob_3$, i.e.

```
1  <https://logRT.org/logevents/e5> a l2tap:PrivacyEvent;
2    l2tap:memebrOf <https://logRT.org>;
3    l2tap:eventParticipant<https://logRT.org/participants/PN_ACLAgent>;
4    l2tap:eventTimestamp "2014-01-30T19:01:00Z"^^xsd:dateTime;
5    l2tap:publicationTimestamp "2014-01-30T19:01:01Z"^^xsd:dateTime;
6    l2tap:eventData <https://logRT.org/logng/ng4>.
7  <https://logRT.org/logng/ng4> = {
8  <https://RT.org/responses/res1> a scip:AccessResponse;
9    scip:responseTo <https://RT.org/requests/req1>;
10   scip:accessDecision "True"^^xsd:boolean;
11   scip:contextObligation <https://logRT.org/req1/obs/ob2>;
12   scip:contextObligation <https://logRT.org/req1/obs/ob3>;
13   scip:propositionalExpression <https://logRT.org/exp/phy1> .
14 <https://RT.org/req1/obs/ob2> a scip:Obligation;
15   scip:createdFrom <https://RT.org/obs/ob2>.
16 <https://RT.org/req1/obs/ob3> a scip:Obligation;
17   scip:createdFrom <https://RT.org/obs/ob3>.}
```

Figure 7: Log event for access response

all three obligations must be fulfilled for the access to be compliant. The `scip:propositionalExpression` property in line 13 encodes this formula. The rest of the quads in Fig. 7 links each of the performable obligations to the corresponding obligation templates in the privacy policy. So the characteristics of each obligation such as the action and the temporal constraints associated with the obligation become resolvable.

The *who* in this log event (line 3) is `https://logRT.org/participants/PN_ACLAgent` indicating that the participant who has logged the response is an ACL agent of PhysioNet, implementing access control and obligation derivation. These mechanisms are usually domain-dependent. In [23] we described how we can derive obligations from privacy preferences using SPARQL queries. Obligations also can be derived using more-or-less complex mechanisms. From the logging perspective what is necessary is to have a mechanism in place to log the access decisions and obligations, regardless of which mechanism is used to control access or derive obligations.

When the obligations are derived from the privacy preferences (obligation templates) and logged, the obligation performer who can be the same participant as the data requestor (Mark, the researcher) or a different participant must fulfill the obligation in an acceptable time interval and log its fulfillment. SCIP has a number of properties to capture the participant who should perform an obligation (`scip:obligationPerformer`), the participant who actually performs the obligation (`scip:performedBy`), the one who can witness the violation of an obligation (`scip:obligationWitness`) and the one who actually witnesses (`scip:attestsViolation`).

Obligation Acceptance. When the access response has been logged, the research team (as the obligation performer) accepts to perform the obligations. This event captures the researcher's commitment as a performative act. The performative act is the utterance of a self-describing act which is performed by declaring that one is doing it [1]. Fig. 8 shows the log event for obligation acceptance. The event's participant (line 3) is Mark, one of the registered researchers. This event refers to the URI of the access response (line 9). By the virtue of logging this event, the researcher not only acknowledges existence of the obligations but also as a performative act commits himself to perform the obligations as conditions to access.

```
1  <https://logRT.org/logevents/e61> a l2tap:PrivacyEvent;
2    l2tap:memebrOf <https://logRT.org>;
3    l2tap:eventParticipant <https://RT.org/ClinicalResearchers/Mark>;
4    l2tap:eventTimestamp "2014-01-31T12:01:00Z"^^xsd:dateTime;
5    l2tap:publicationTimestamp "2014-01-31T12:01:01Z"^^xsd:dateTime;
6    l2tap:eventData <https://logRT.org/logng/ng9>.
7  <https://logRT.org/logng/ng9> = {
8  <https://RT.org/acceptances/acpt1> a scip:ObligationAcceptance;
9    scip:accepts <https://RT.org/responses/res1>.}
```

Figure 8: Log event for obligation acceptance

```
1  <https://logRT.org/logevents/e6> a l2tap:PrivacyEvent;
2    l2tap:memebrOf <https://logRT.org>;
3    l2tap:eventParticipant <https://RT.org/ClinicalResearchers/Mark>;
4    l2tap:eventTimestamp "2014-02-01T12:01:00Z"^^xsd:dateTime;
5    l2tap:publicationTimestamp "2014-02-01T12:01:01Z"^^xsd:dateTime;
6    l2tap:eventData <https://logRT.org/logng/ng5> .
7  <https://logRT.org/logng/ng5> = {
8  <https://RT.org/req1/performedobs/ob2> a scip:PerformedObligation;
9    scip:performedFor <https://RT.org/req1/obs/ob2>;
10   scip:performedBy <https://RT.org/ClinicalResearchers/Mark>;
11   scip:occurredIn "2014-02-01T11:00:00Z"^^xsd:dateTime .}
```

Figure 9: Log event for performing an obligation

Performing Obligation. In the scenario, one of the research team members (Mark) is the participant who must perform the obligations as conditions to access to the dataset. The first obligation (`obtain_training_certificate`) is a pre-obligation, meaning that the research team must obtain the certificate and log this action as an evidence prior to access. Fig. 9 shows a log event that captures the fact that Mark has performed the first obligation. Line 8 defines the performed obligation as an instance of `scip:PerformedObligation` class. Line 9 refers to the URI of the corresponding obligation logged in the access response. The participant who has performed the obligation and the time instant of performing the obligation are encoded using the `scip:performedBy` (line 10) and `scip:occurredIn` (line 11) respectively. Note that Mark is the *who* has submitted these quads to the log (line 3).

Access activity. Finally, SCIP has a class `scip:AccessActivity` to record the occurrence of an access activity. Fig. 11 shows the log event of an access activity when the research team (including all its members) has accessed the dataset. Line 9 refers to the URI of the corresponding obligation acceptance event using `scip:forObligationAcceptance`. Line 10 captures the time instant that the access activity occurred. The provenance assertions for this log event (line 3) shows that the researcher is the participant who logs the access activities. We assume that the data provider (PhysioNet) has also a mechanism in place to log all accesses to its dataset. Therefore, if the researcher fails to log an access activity the discrepancy between the provider's access log and the L2TAP audit log will trigger a non-compliance incident.

The justification for leveraging the Linked Data infrastructure and derferenceable URIs become evident as we walk through the log events for the motivating scenario described above. We summarized registration of the log events in Fig. 11. Participants make statements about the events in the log. Therefore, they need to access the events data to dereference the past events URIs that may have been

```
<https://logRT.org/logevents/e7> a l2tap:PrivacyEvent;
  l2tap:memebrOf <https://logRT.org>;
  l2tap:eventParticipant<https://RT.org/ClinicalResearchers/Mark>;
  l2tap:eventTimestamp "2014-02-02T00:01:00Z"^^xsd:dateTime;
  l2tap:publicationTimestamp "2014-02-02T00:01:01Z"^^xsd:dateTime;
  l2tap:eventData <https://logRT.org/logng/ng6> .
<https://logRT.org/logng/ng6> = {
<https://RT.org/access/req1/ac1> a scip:AccessActivity;
  scip:forObligationAcceptance <https://RT.org/acceptances/acpt1>;
  scip:occurredIn "2014-02-02T00:00:01Z"^^xsd:dateTime .}
```

Figure 10: Log event for access activity

logged by other participants in the different points in time. For example, the privacy policies are registered by PhysioNet on Jan 01, 2014, then the access request has been logged by the research teams on Jan 29, 2014. The access response event has been logged by the Physionet access control agent on Jan 30th referring the URI of the access request (`scip:responseTo <https://RT.org/requests/req1>`). The access response also refers to the URIs of obligations registered by PhysioNet as part of the privacy policies (e.g. `scip:createdFrom <https://RT.org/obs/ob2>`). Analogously the log events encoding the acceptance of obligations, fulfilment of an obligation by one of the researchers and the access activity logged by the research team refer to the URIs of the other past log events. The statements that each of these participants wants to make depends on the URIs of the statements have been previously logged.

The events in Fig. 11 do not necessarily occur in the sequence shown. Consider a scenario in which the researcher logs an access to the dataset referencing an obligation acceptance's URI. However, the researcher happens to not log an obligation fulfilment event corresponding to the access response. So the access response's URI not be referred by a performed obligation event and in turn the corresponding access request would also not be referred. This results in a non-compliant access request and the researcher would become accountable for not logging the obligation fulfilment event. Therefore, the L2TAP+SCIP ontology relies on the URI dereferencing to make actions of each participant transparent for other participants involved in the process (of course for the participants who have been authenticated) and provide support for accountability and privacy.

3. QUERY-BASED AUDITING

The fundamental aspect of leveraging RDFS and Linked Data to generate L2TAP logs is to facilitate privacy audit tasks by queries over the created logs. In this section we will first discuss how the standard RDFS and computation of transitive closures for the `refs:subClassOf` relationship can be exploited to support query-bases audit tasks. Then we describe three major audit tasks (*constructing the log with data usage policies*, *obligation derivation and fulfilment*, and *compliance checking*) that all can be supported by SPARQL queries with a limited RDFS reasoning support. These tasks involve several classes of participants including data provider, data receiver (research teams), and auditors.

RDFS Reasoning Support. By leveraging Linked Data for privacy audit log we can achieve a flexible way to deal with data items granularity, participants granularity, and applicable privacy policies. An individual's personal information can span from a very specific data item (e.g. the glucose level in a blood work) to a very general data item (e.g. the personal health record of an individual). Privacy policies and regulations (e.g. HIPPA) are not only applicable to an entire dataset but also may apply to a specific class of data items (e.g. mental health data) or a specific class of individuals (e.g. children under age of 12). Individuals (e.g. data subjects in our scenario) may have options to express their personal privacy preferences applicable to the instances of their data. Expressing everything in the log (including data items, participants, etc.) using dereferencable URIs provides the most flexible and generic way of representing resources involved in privacy processes. Furthermore, RDF representation of audit logs using L2TAP and SCIP ontologies allows both the URI of a class of resources or URI of an instance of a resource (participants or data items) to be dereferenced and reasoned about using RDFS. As shown in Fig. 3 members of the class of researchers are defined using a named graph as a log event payload. Exploiting `rdfs:subClassOf` allows to reason about the entire class of researchers or a specific individual in the class when evaluating an obligation derivation query or a compliance query as described below. With the same token applicable privacy policies and preferences can be determined for a class of data subjects, a class of data items, or for one instance of the same classes.

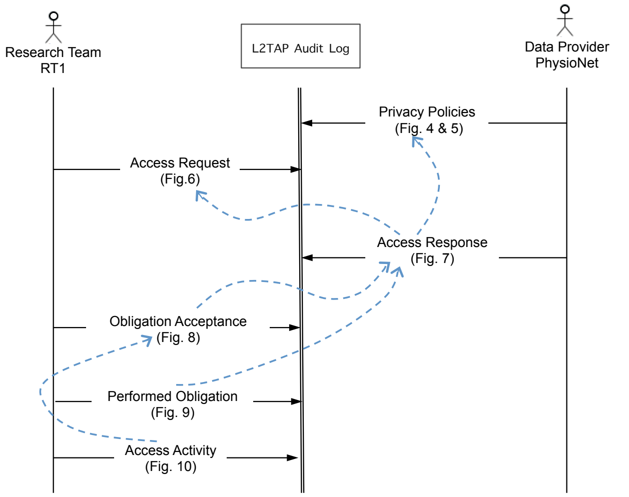

Figure 11: Registering log events into the log and required URI dereferencing

Log Construction. In our motivating scenario, the research institute is the one who needs access to the datasets for its researchers and also wants to keep its researchers accountable with respect to the dataset usage policy. Therefore, the research institute initializes the log and registers the participants. The institute then uses the log in the future and show to the interested auditors that its researchers are compliant with the policies. On the other hand, the data provider wants to be able to express the norms and policies that govern the data usage. So the provider wants to contribute to the log these policies and record all accesses to datasets. We illustrated throughout Fig. 1-4 the set of quads that need to be stored in an L2TAP log for these tasks.

All quads in these figures can be appended to an L2TAP log using SPARQL 1.1 [29] commands in three steps: first a named graph will be created for a log event using `CREATE GRAPH <g>`, second the quads of the log header will be inserted to the log default graph and then the quads of the log event body will be inserted into the named graph using `INSERT DATA {GRAPH <g> { }}`.

Obligation Derivation. After the log is constructed, the research teams (or an individual researcher) want to be able to derive the obligations applicable to the class of data items or data subjects that they want to access. This task can be accomplished through computation of transitive closures for the `rdfs:subClassOf` relationship. Norms in the SCIP ontology are defined in terms of data items, roles of participants who want to use data items, purpose of usage, and requested access privilege. All these concepts are expressed in SCIP by a lattice using `rdfs:subClassOf`. For example children under 12 are `rdfs:subClassOf` data subjects. Therefore, a SPARQL query with the RDFS reasoning support allows to match the context of a set of privacy policies with the context of an access request. The query conditions check that all instances of data items, data subjects, roles, privacy privileges asked by the research teams in the access request graph, can be subsumed by the corresponding items in the privacy policies graph. Then the output of the query will be applicable obligations to that access request. The method has been described in more details in our earlier publication ([23]-Section 3).

Compliance Checking. An important audit task is to identify, at any given point in time, if an access request is in compliance with the applicable privacy policies. Compliance of an access request is decided based on the status of its corresponding obligations. Therefore, a typical compliance checking task will be performed in three steps as illustrated in Algorithm 1. First multiple SPARQL ASK queries evaluate the status of all individual obligation and return true for an obligation if it is fulfilled and false otherwise. The template query shown in Fig. 12 can be used for evaluating the fulfilment of an obligation after the parameter `@ob` is substituted with the URI of an obligation. A similar template query can be used to evaluate a pending obligation (an obligation that the conditions for its fulfillment not yet settled).

For each access response a propositional formula will be also logged indicating how the fulfilment of an individual obligation contributes to the overall compliance of an access request. In our scenario the formula is $\varphi \equiv ob_1 \wedge ob_2 \wedge ob_3$ i.e. all three obligations must be fulfilled for the access request to be compliant. The second step in the algorithm is to substitute the propositional variable in φ with the truth-values representing the state of every derived obligation. Each ob_i in this formula will be substituted with o_i which can be true or false depending on the evaluation of the query in Fig. 12.

```
ASK
  WHERE {
    ?obAcc scip:accepts ?response.
    ?response scip:responseTo ?request.
    ?response scip:contextObligation @ob.
    @ob rdf:type scip:Obligation.
    @ob scip:occurrenceGap ?occGap.
    @ob scip:performanceDuration ?pD.
    OPTIONAL {?accessActivity scip:forObligationAcceptance ?obAcc}.
    OPTIONAL {?accessActivity scip:accessedTime ?accessTime}.
    OPTIONAL {?performedOb scip:performedFor @ob}.
    OPTIONAL {?performedOb scip:performedBy ?performAgent}.
    OPTIONAL {?performedOb scip:occurredIn ?obligationTime}.
    OPTIONAL {?witness scip:attestsViolation @ob}.
    FILTER (((!bound(?performAgent) && !bound (?accessTime))
    ||(bound (?accessTime) && (xsd:integer(@currentTime) < =
    fn:max((xsd:integer(?accessTime) + xsd:integer(?occGap) + xsd:
          integer (?pD)),
    (xsd:integer(?accessTime) + xsd:integer(?occGap)))))) &&
    (!bound(?witness)))  }
```

Figure 12: Evaluating the fulfilment of an individual obligation

```
  Data: Access request: $rq$, currentTime: $t$
  Result: Boolean Compliance value for $rq$
1 $OB \leftarrow$ set of derived obligations for $rq$ ;
2 $\phi \leftarrow$ propositional formula for $rq$;
3 foreach $ob_i \in OB$ do
4     $o_i \leftarrow answer\_of$ (SPARQL ASK obligation query (Fig. 12));
5     Substitute $ob_i$ in $\phi$ with $o_i$ ;
6 end
7 Substitute $\phi$ in Compliance Ask Query;
8 $C \leftarrow answer\_of$ (SPARQL ASK compliance query (Fig. 13));
9 return ($C$)
```

Algorithm 1: An algorithm for compliance checking

The third step in the algorithm is to substitute φ as a propositional variable and evaluate the template query in Fig. 13 to check the overall compliance of the corresponding access request. Note that in line 3 of the query in Fig. 13, we include the graph encoding the access decision of the access request. The `?accessDecision` variable is a propositional variable that will be used in the expression in line 4 to evaluate the access request compliance queries. The `FILTER` statement is the conjunction of φ and `?accessDecision` meaning that if the access decision logged by the access control mechanism is false even if all obligations are fulfilled the access request would be non-compliant.

```
1 ASK
2 WHERE { ?response scip:responseTo @rq .
3   ?response scip:accessDecision ?accessDecision .
4   FILTER (@phi && xsd:boolean(?accessDecision)) }
```

Figure 13: Evaluating an access request compliance

A number of other compliance queries (e.g. which obligation is pending or which access request is not compliant at time t), the experimental validation of the scalability of our solution, and the practical benefits of our approach are described in [23].

4. RELATED WORK

Our research study is inspired by the concept of *information accountability* as described by Witzner et al. [30], that is ensuring whether the policies and configured preferences that govern the flow of personal information, are respected by the parties that collect, use, and share users' data. In an early work on the management of policies and the semantic web [14], Kolovski et al. emphasize on the need for a declarative access policies to support scalable information sharing among parties. The authors then propose a rule-based discretionary access control language for the web. In [13], Kagal et al. propose Rein, a policy framework grounded in semantic web technologies. The authors acknowledge and respect the diversity and heterogeneity of policy languages on the web and propose Rein as an ontological framework for policy interoperability. The ontology proposed in this paper supports information accountability via privacy audit logs and complements the Rein proposal [13] by providing a SPARQL query based solutions for the basic compliance checking queries.

There are solid theoretical foundations for policy auditing over logs [2, 9, 3, 5]. Barth et al. use Alternating-time Temporal Logic to build a logical privacy model and design a privacy language (LPU) to express norms [2]. The concept of norms in this work has been adapted from the Contextual Integrity perspective [20]. The LPU language allows all communications between agents to be recorded in a logical trace. Norms are expressed as logical constraints and privacy compliance is related to the logical concepts of satisfiability and entailment. Datta et al. [9] extended the LPU language with reasoning about information accountability over incomplete logs. Basin et al. use metric first order temporal logic (MFOTL) to express policies, which are then monitored to verify whether the trace of actions satisfies desired temporal properties [3]. Cederquist et al. describe a framework that uses audit logs to enforce compliance with discretionary access control policies [5]. While this body of work propose highly expressive privacy logic, lack of support by an scalable semantic technology prevents the approaches to be applied outside of research labs.

An important related work is the recently proposed RDF provenance model (PROV-DM) [17]. The focus of PROV-DM is on providing a domain independent ontology for asserting provenance of a resource on the web. While the provenance assertions of the L2TAP+SCIP log events (log event header) can be expressed using PROV-DM ontology, the ontology cannot support the structure needed to encode the semantics of the body of privacy events (e.g. privacy preferences, obligations, and purpose of usage). A simple mapping between L2TAP and PROV-DM allows a log event (regardless of its content) to be expressed by the PROV-DM ontology. The mapping requires adding a `prov:Activity` (i.e. defining a URI for the act of generating the `l2tap:LogEvent` as a `prov:Entity`). Then the assertion of the *who*, `l2tap:eventParticipant`, will be mapped to the `prov:wasAssociatedWith` property. The two L2TAP properties capturing the *when* assertions are mapped to `prov:startedAtTime` and `prov:endedAtTime` respectively.

In recent years, we have seen several proposals addressing privacy in the Linked Data context ([22, 18, 7, 8]). This body of research are mainly proposing access control frameworks based on access control lists (ACLs). Authors in [22] propose a privacy preferences vocabulary that can be utilized to express fine-grained access policies in Linked Data environment. Muhleisen et al. propose an access control mechanism for social web applications [18]. This framework uses SWRL to express access rules. Authors in [12, 7, 8] leverage the Linked Data architecture for providing authorizations and access restrictions at the document level [12]. The authorization mechanism in [12] is based on WebID [25]. To address the privacy concerns in the emerging domains of linked data applications, Speiser et al. [24] propose a privacy framework for policy specification and access control enforcement.While access control is a necessary mechanism to protect individuals' privacy, it is not sufficient to express and control data usage policies. The work introduced in this paper addresses privacy concepts such as usage purposes and obligations after access.

5. CONCLUSIONS

While compliance auditing is mandated in different privacy legislation (e.g. [26, 21]), it has received less attention from the research community. In this paper we continued our work in [23] and showed that regardless of what logic is used to express privacy policies there is a standard way for privacy logging that allows basic privacy events to be logged and provides a scalable query-based solution for answering compliance queries. We also demonstrated that L2TAP Linked Data Log is capable of facilitating basic privacy auditing tasks such as: constructing the log, obligation derivation, and compliance checking in the big data and linked data research context. In our approach, the convenience of Linked Data and RDFS has been sought for privacy log interoperability and facilitating accountability and transparency among participants.

6. ACKNOWLEDGMENTS

Financial supports from the NSERC Canada and Privacy Awards from IBM and the Information and Privacy Commissioner of Ontario are greatly acknowledged.We thank the anonymous reviewers for their comments.

7. REFERENCES

[1] J. Austin. *How to do things with words*, volume 88. Harvard University Press, 1975.

[2] A. Barth, A. Datta, J. C. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: Framework and applications. In *Proc. SP*, pages 184–198, 2006.

[3] D. Basin, F. Klaedtke, and S. Müller. Policy monitoring in first-order temporal logic. In *Proc. CAV*, pages 1–18, 2010.

[4] J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(4), 2011.

[5] J. Cederquist, R. Corin, M. Dekker, S. Etalle, J. den Hartog, and G. Lenzini. Audit-based compliance control. *Int. J. of Info. Security*, 6:133–151, 2007.

[6] A. Chuvakin, E. Fitzgerald, R. Marty, R. Gula, W. Heinbockel, and R. McQuaid. Common event expression, 2008.

[7] L. Costabello, S. Villata, N. Delaforge, F. Gandon, et al. Linked data access goes mobile: Context-aware authorization for graph stores. In *LDOW- WWW*, 2012.

[8] L. Costabello, S. Villata, O. R. Rocha, and F. Gandon. Access control for http operations on linked data. In *ESWC*, pages 185–199, 2013.

[9] A. Datta, J. Blocki, N. Christin, H. DeYoung, D. Garg, L. Jia, D. Kaynar, and A. Sinha. Understanding and protecting privacy: formal semantics and principled audit mechanisms. In *Proc. ICISS*, pages 1–27, 2011.

[10] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus,

G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[11] T. Heath and C. Bizer. Linked data: Evolving the web into a global data space. *Synthesis Lectures on the Semantic Web: Theory and Tech.*, 1(1):1–136, 2011.

[12] J. Hollenbach, J. Presbrey, and T. Berners-Lee. Using RDF metadata to enable access control on the social Semantic Web. In *Proc. CCMLSK WS at CK*, 2009.

[13] L. Kagal, T. Berners-Lee, D. Connolly, and D. Weitzner. Using semantic web technologies for policy management on the web. In *Proc of the National Conference on Artificial Intelligence*, 2006.

[14] V. Kolovski, Y. Katz, J. Hendler, D. Weitzner, and T. Berners-Lee. Towards a policy-aware web. In *Semantic Web and Policy Workshop at the ISWC*, 2005.

[15] S. Loosemore, R. Stallman, R. McGrath, A. Oram, and U. Drepper. *The GNU C library reference manual*. Free software foundation, 2001.

[16] G. B. Moody and L. Lehman. Predicting acute hypotensive episodes: The 10th annual physionet/computers in cardiology challenge. In *Computers in Cardiology*, pages 541–544. IEEE, 2009.

[17] L. Moreau and P. Missier. PROV-DM: The PROV data model. W3C Recomm., W3C, June 2012.

[18] H. Mühleisen, M. Kost, and J.-C. Freytag. SWRL-based Access Policies for Linked Data. In *Proc. SPOT Workshop at SSW*, 2010.

[19] Q. Ni, E. Bertino, and J. Lobo. An obligation model bridging access control policies and privacy policies. In *Proc. SACMAT*, pages 133–142, 2008.

[20] H. Nissenbaum. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford Law Books, 2009.

[21] Official Journal of the EC. *EU directive 95/46/EC on the protection of individuals rights with regard to the processing of personal data*, 1995.

[22] O. Sacco and A. Passant. A privacy preference ontology (PPO) for Linked Data. In *Proc. LDOW, WWW*, 2011.

[23] R. Samavi and M. P. Consens. L2TAP+SCIP: An audit-based privacy framework leveraging Linked Data. In *CollaborateCom (TrustCol)*, pages 719–726, 2012.

[24] S. Speiser. Policy of composition? composition of policies. In *Proc. POLICY*, pages 121 –124, 2011.

[25] H. Story, B. Harbulot, I. Jacobi, and M. Jones. FOAF+SSL: RESTful Authentication for the Social Web. In *Proc. SPOT*, 2009.

[26] US Congress. *Health Insurance Portability and Accountability Act of 1996, Privacy Rule. 45 CFR 164*, Aug. 2002.

[27] US Department of Health and Human Services. *Code of Federal Regulations, Title 45 - Part 46 - Protection of Human Subject*, Revised January 15, 2009.

[28] W3C. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C, April 2002.

[29] W3C. *SPARQL 1.1 Query Language, W3C Proposed Recommendation*. W3C, November 2012.

[30] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G. J. Sussman. Information accountability. *Commun. ACM*, 51(6):82–87, 2008.