

LIoPY: A legal compliant ontology to preserve privacy for the Internet of Things

Faiza Loukil, Chirine Ghedira, Khouloud Boukadi, Aïcha-Nabila Benharkat

▶ **To cite this version:**

Faiza Loukil, Chirine Ghedira, Khouloud Boukadi, Aïcha-Nabila Benharkat. LIoPY: A legal compliant ontology to preserve privacy for the Internet of Things. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Jul 2018, Tokyo, Japan. 10.1109/COMPSAC.2018.10322 . hal-01825750

HAL Id: hal-01825750
https://hal.science/hal-01825750

Submitted on 21 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LIoPY: A legal compliant ontology to preserve privacy for the Internet of Things

Faiza Loukil *, Chirine Ghedira-Guegan ‡, Khouloud Boukadi §, and Aïcha Nabila Benharkat ¶
*University of Lyon, University Jean Moulin Lyon 3, CNRS, LIRIS, FRANCE; Email: faiza.loukil@liris.cnrs.fr
‡University of Lyon, University Jean Moulin Lyon 3, iaelyon school of Management,
CNRS, LIRIS, FRANCE; Email: chirine.ghedira-guegan@liris.cnrs.fr
§University of Sfax, Mir@cl Laboratory, Sfax, TUNISIA; Email: khouloud.boukadi@fsegs.usf.tn
¶University of Lyon, INSALyon, CNRS, LIRIS, FRANCE; Email: nabila.benharkat@liris.cnrs.fr

***Abstract*—The Internet of Things (IoT) provides the opportunity to collect, process and analyze data. This opportunity helps to understand preferences and life patterns of individuals in order to offer them customized services. However, privacy has become a significant issue due to the personal nature of the knowledge derived from these data and the involved potential risks. Despite the increasing legislation pressure, few proposed solutions have dealt with the privacy requirements, such as consent and choice, purpose specification, and collection limitation. In this paper, we propose a privacy ontology in order to incorporate privacy legislation into privacy policies while considering several privacy requirements. Our proposed ontology aims both at making the smart devices more autonomous and able to infer data access rights and enforcing the privacy policy compliance at the execution level. We implemented and evaluated our privacy ontology based on a healthcare scenario.**

I. Introduction

IoT connects and shares data collected from smart devices, such as medical and household appliances. By 2020, the number of connected devices is expected to reach 50 Billion [2]. This rise represents an opportunity for the IoT to improve the efficiency and the quality of life and provide everyday conveniences to the users [7].

Usually, IoT requires the collaboration of several parties to perform a common task. For instance, medical wearable devices have been developed to sense and collect individuals' vital signs. Those latter can be remotely monitored through applications. However, the collected and shared data are full of sensitive data. Therefore, the data owner wishes not to share them with all these heterogeneous organizations without retaining some level of control.

The privacy violation risks and the legislation pressure force both the data owner and the requester to preserve the privacy of the shared data. However, matching the data owner's preferences and the requester's access query requires the use of the same privacy vocabulary that describes the privacy requirements. This matching enables the creation of a common privacy policy that can be applied to preserve the data owner privacy in the IoT environment while handling his data.

To express such privacy vocabulary, we propose to define a European legal compliant ontology to preserve privacy for IoT, called LIoPY. Our ontology will identify and define the basic concepts for the description of privacy requirements. Moreover, the data owner has often not enough experience and expertise in the privacy domain to take advantage of his legal rights [10]. For this reason, semantic modeling becomes fundamental to infer the required privacy obligations that are added to the privacy policy to be performed on the shared data.

Privacy of personal information, as defined by the NISTIR [12], involves the right to control when, where, how, to whom, and to what extent an individual shares his own personal information, as well as the right to access personal information given to others, to correct it. All these privacy requirements exist in the European regulation [10] and privacy standards [5] [9]. In our work, 'privacy requirements' means the obligations that must be fulfilled by all the involved parties, such as the data owner, requester, and cloud storage service in the process of personal data treatment to preserve the privacy during the whole data lifecycle, covering the collection, transmission, storage and processing phases. To our knowledge, existing solutions, which propose ontology-based privacy preserving in IoT focus only on who can access the data collected by smart devices and usually do not address the whole data lifecycle. However, addressing only access control at processing time is not enough to preserve privacy [10].

Motivated by the above legal rights and missing works, we focus on the privacy requirements to preserve privacy driven by legislation [10] and standard [5] during the whole data lifecycle. We look at how to incorporate these privacy requirements and obligations into a privacy end-to-end model for the IoT environment. To this end, we propose the definition of a privacy ontology, called LIoPY combined with standard reasoning technologies to address the privacy requirement compliance from an end-to-end view. In fact, our proposal involves the right of each owner to control and keep the data ownership once his data are shared by defining his privacy preferences related to the collection, transmission, storage and use phases.

This paper is organized as follows. Section 2 gives an overview of the literature and discusses the ontology-based solutions for preserving privacy in the IoT applications. Section 3 describes our proposed ontology, called LIoPY to preserve privacy by detailing the main LIoPY modules. The system architecture is then presented in Section 4. Section 5 experiments our LIoPY on a healthcare scenario. Section 6 concludes the paper and presents some future endeavors.

II. Related Work

Current research approaches used the expressive power of semantics and ontologies to preserve privacy in the IoT environment. We can distinguish between two categories, namely (i) ontology-based IoT resource description [1] [3] [6] [14] that focused on illustrating the IoT resources but omitting reasoning over privacy policies and (ii) privacy-aware access control [4] that focused only on who can access the generated data by the IoT resources at processing time.

Bermudez et al. [1] proposed an instantiation of the semantic sensor network (SSN) ontology [3], called IoT-Lite that describe key IoT concepts allowing interoperability and discovery of sensed data in heterogeneous IoT platforms by a lightweight semantics. Kotis and Katasonov [6] proposed an ontology to present the abstraction of smart entities, which are sensing/actuating devices that observe some features of interest or act on some other entities, and control entities, which are applications that use the sensed data from the smart entities. The proposed ontology aimed at supporting the automated deployment of control entities in settings where smart entities have been already deployed. The above-mentioned schemes achieved the modeling establishment and initial reasoning, but did not take into account the IoT privacy issue.

Wang et al. [14] proposed an Ontology-based Resource Description Model (ORDM) that describes resources in the IoT environment, which are described by the attribute, state, control, historical information and privacy classes. The Attribute class defines the inherent information of the device, such as the device type, model, and range of the sensed values. The data description is made in the State class, which provides the current data captured by the sensor. The Privacy class protects the device from illegal access or control. However, ORDM did not offer a fine-grained access control to the sensed data. Indeed, the users that can access the IoT resource are fixed in this ontology without any reasoning or clear criteria. Moreover, the authors did not deal with data resource sharing during the data processing phase. Furthermore, only access and control authorities are considered as privacy requirements. ORDM did not consider the rest of privacy requirements, such as purpose specification, retention, and disclosure limitation.

Hosseinzadeh et al. [4] proposed a context-aware, role-based access control model for smart spaces in the healthcare domain. The access control scheme is modeled using ontological techniques and the Web Ontology Language (OWL). It supports automatic data reasoning and inferring. There are two sets of rules, namely rules designed by the administrator, and rules defined by the user to protect personal data. A central access control service makes all decisions about the access rights. However, the authors did not deal with the whole data lifecycle. They just focus on the processing data phase. Besides, some privacy requirements, such as patient's consent, retention, and disclosure limitation are ignored in this work.

To sum up, it can be said that the existing solutions concerning ontology-based for preserving privacy in IoT did not address the whole data lifecycle. Moreover, they focus only on who can access the data sensed by smart devices. Despite the progress made by the discussed solutions, it seems necessary to propose a novel approach that covers the whole data lifecycle and preserves privacy by involving the different privacy requirements that are cited in legislation [10] and privacy standards [5] [9]. Moreover, smart devices need to become autonomous by giving them the capability to infer access rights according to the owner's privacy preferences. Thus, we propose LIoPY, a European legal compliant ontology for IoT privacy-preserving that provides a solution for overcoming the aforementioned existing model limitations.

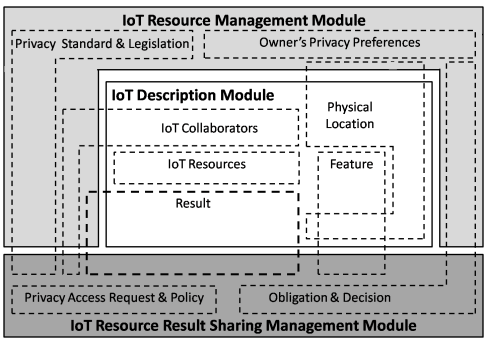


Fig. 1: Modules in the legal IoT privacy ontology (LIoPY)

III. Overview of the Legal IoT Privacy Ontology(LIoPY)

IoT is an open and dynamic environment where smart devices collaborate with one another to perform high-level tasks. We choose to describe the IoT data using the ontology in order to abstract the implementation issues and to identify and define the basic concepts for the description of privacy aspects in a domain-independent manner. Moreover, the use of ontology enables us to resolve the interoperability issue on different concepts used by different IoT actors and resources across heterogeneous domains to define the shared data.

The main purpose of proposing our ontology is to protect privacy during the whole process of collecting, transmitting, storing, and processing the collected data by smart devices. We use standard ontology languages to define a common privacy vocabulary combined with standard reasoning technologies based on description logic to address the privacy requirements.

Moreover, in order to be as interoperable as possible, LIoPY imports the Semantic Sensor Network (SSN) ontology [3] to point out to some classes and extend it with some privacy and security properties. SSN is published and recommended by the World Wide Web Consortium (W3C). SSN includes a lightweight but self-contained core ontology called SOSA (Sensor, Observation, Sample, and Actuator). Although the SSN ontology presents knowledge in the domain of sensor networks, it lacks the relevant definitions to preserve privacy.

Ontology modularization is a common method used in ontology engineering to segment an ontology into smaller parts. Figure 1 shows an overview of LIoPY. In order to cover the whole privacy aspects, LIoPY contains three main modules,

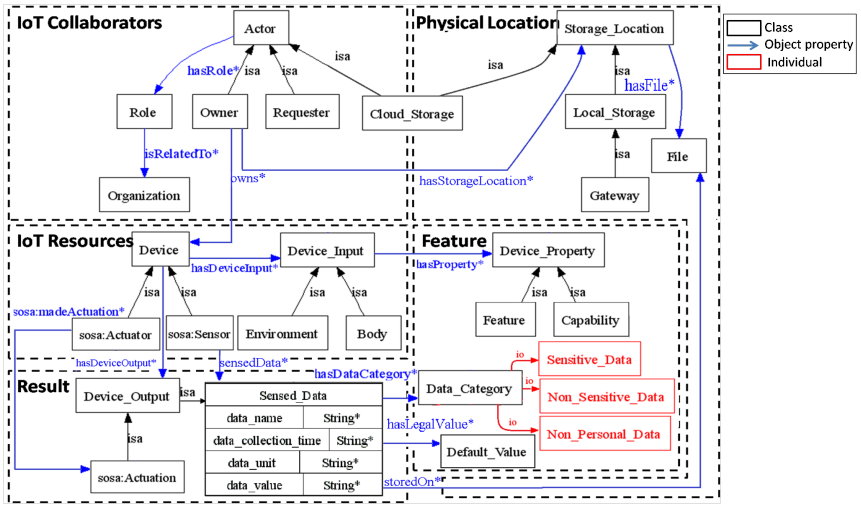


Fig. 2: IoT Description Module in LIoPY

namely IoT description, IoT resource management, and IoT resource result sharing management. Each module includes a set of sub-modules. It aims at providing the LIoPY users with the knowledge they require, reducing the scope as much as possible to what is strictly necessary in a given use case.

A. IoT Description Module

The IoT Description Module describes the IoT environment. It includes five sub-modules, namely (i) IoT Collaborators, which includes the different involved parties that collaborate together in order to achieve a common goal in the IoT domain, (ii) IoT Resources, which focus on modeling sensors and sensor network, (iii) Result, which describes the output produced by the IoT resources, (iv) Physical Location, which includes the storage location of the IoT resource results and data owner's preferences, and (v) Feature, which describes the features of both the IoT resources and their results. Figure 2 shows the defined classes of the IoT description module and the relationships among them. We reuse some concepts of the SSN Ontology [3] and extend it with a new class (Sensed_Data class) in order to facilitate the access, use, and verification of the information generated by IoT resources.

We will detail below the second module, which aims at preserving privacy before and during the collection phase.

B. IoT Resource Management Module

The IoT Resource Management module helps the owner to better control his IoT resources and their results. Hence, this module includes two sub-modules, namely:

1) Privacy Standard and Legislation sub-module: The Organization for Economic Co-operation and Development (OECD) [9] and the ISO standard [5] provides several principles that enable individuals to express their privacy requirements and place obligations on organizations to follow. Moreover, the European Regulation [10] embed these principles.

Based on the NISTIR privacy definition [12] and the studied principles, we define the Privacy_Attribute class. Table I presents the Privacy_Attribute subclasses.

TABLE I: Privacy attribute subclasses based on [10] [12]

Privacy Attribute Subclass	Description
Purpose	Reasons for requesting data
Retention	Time duration for storing data
Operation	Allowed usage on the shared data
Disclosure	Explicit data owner acceptance for sharing data with third parties
Condition	Required circumstance, such as data owner age, technical capabilities, etc.

2) Owner's Privacy Preferences sub-module: This sub-module helps the owner to define the rules to preserve the privacy of his collected data and to express his privacy preferences on how his smart devices must behave. It includes both the Privacy_Rule and Privacy_Permission_Setting classes.

a) Privacy_Rule Class: It specifies how the shared data must be handled by requesters. To this end, one or more rules can be associated with each device output using the object property hasPrivacyRule.

We represent the Privacy_Rule class, denoted as $PRule$, as a tuple of the following form: $PRule =< cons, pAtt, obl >$ The data owner consent, denoted as **cons**, is an element of the predefined set of individuals of the Consent class, denoted as CONS with $cons \in CONS$ and CONS=(NeedForConsent, NoNeedForConsent). It specifies if an explicit data owner's consent is mandatory or not in order to apply the privacy rule.

Moreover, for each privacy rule and by using the hasPrivacyAttribute, several privacy attributes **pAtt** $\in PATT$, can be defined to preserve data owner's privacy, with $pAtt = \{purpose, retention, disclosure, operation, condition\}$. Each attribute value belongs to the set of its predefined values. For example, the purpose attribute belongs to the set of individuals of the Purpose class: **purp** $\in$ {Statistical, Research, Treatment}.

Furthermore, a privacy obligation, denoted as **obl**, can be associated with each privacy rule.

b) Privacy_Permission_Setting Class: Privacy of the person, as defined by NISTIR [12], is the right to control the

integrity of his body. It covers physical requirements, health problems, and required medical devices. In order to guarantee this right, our ontology enables the definition of some settings to manage smart devices and their generated results. To this end, we propose the Privacy_Permission_Setting class that expresses the data owner privacy preferences on how his smart devices must behave according to each Device_Output.

These privacy permission settings are defined by the data owner before the beginning of the collection phase. These permission settings will be locally stored and regularity verified before allowing any device to communicate with other devices or connect to the Internet. The permission setting verification leads to enforce data owner control on his devices and rapidly detect any malicious attempt by analyzing the device behavior.

We will look at the last LIoPY module, which aims at preserving privacy during the rest of the data lifecycle, namely the transmission, storage, and processing phases.

C. IoT Resource Result Sharing Management Module

The IoT Resource Result Sharing Management Module presents how the data must be handled once shared. A set of privacy obligations, such as cryptography, data anonymization, and noise addition should be applied to the data before sharing them with third parties. Indeed, we define the object property hasAccessDecision that have the Privacy_Obligation class as the domain and the Decision class as the range instantiated as Permit or Deny. Moreover, we propose the Access_Request class that uses the same privacy attribute set with the privacy rule to generate a Privacy_Policy that defines how the data can be handled if the access request is accepted. Figure 3 depicts the Privacy_Policy properties.

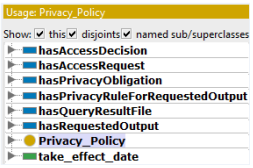


Fig. 3: Privacy_Policy properties

Some relationship cannot be expressed in the OWL. For this, we use the semantic web rule language, SWRL to define a set of inference rules, which are built upon the different LIoPY's concepts and properties. Our reasoning process is based on a set of SWRL rules and ontology classes to infer privacy policies for different possible data sharing cases in the real world. We present below our algorithm and inference rules.

IV. The System Architecture

Figure 4 shows our proposed architecture that enables preserving privacy using our defined LIoPY ontology. The architecture includes three involved parties, namely cloud storage, data requester, and data owner. LIoPY is stored on the cloud storage and shared between all the involved parties. We assume that the cloud storage is a private one. Thus, it offers sufficient security level in order to protect the shared LIoPY against any malicious attempts to alter the ontology

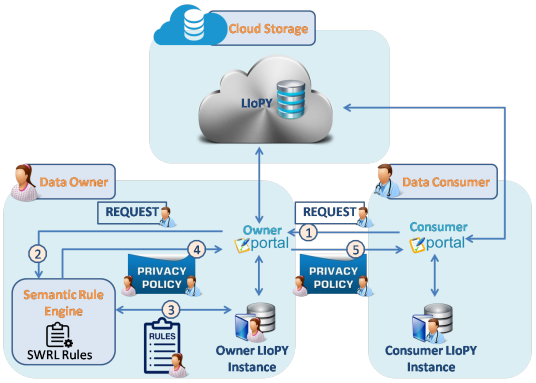


Fig. 4: System Architecture

concepts and/or its inference rules. The data requester uses his portal to create and send his access request to the data owner portal. The data owner uses his portal via his gateway to define his preferences about each device output. The data owner can define two types of preferences, namely permission settings and privacy rules. When an access request is received, the data owner portal communicates with the semantic rule engine, which is responsible for reasoning about the received request and, then taking a decision to create or not a privacy policy. During the reasoning process, the semantic rule engine evaluates the access request by following the described algorithm (see Algorithm 1) using our predefined set of SWRL rules. In case of a match between the access request and the appropriate privacy rules, a privacy policy is created and sent to the data requester portal that will permit any actions on data only if it is allowed in the privacy policy.

A. The reasoning process

Authorization access decisions are taken during the reasoning process to derive the Privacy_Policy. Instantiating a Privacy_Policy class is the result of a successful privacy policy matching between the data owner's rules (Privacy_Rule class) and requester's access request (Access_Request class) using the defined privacy attributes (Privacy_Attribute class). So, we define **Privacy Attribute Matching** Algorithm (see Alg.1).

Algorithm 1 takes as input the data requester's access request and returns the instantiated privacy policy if there is a match. First, the semantic rule engine checks if its related smart devices collect the requested data using its set of device outputs $DeviceOutputs$. If no, it returns an empty privacy policy. Otherwise, it retrieves the appropriate rules. Then, it matches the data owner's privacy rules and the requester's access request to instantiate a privacy policy. In case of a match, a privacy policy is created. It defines how the shared data can be handled by the data requester.

In our work, the data owner can define his data sensitivity and his preferences about how his shared data must be handled by consumers. However, he generally has not enough expertise in the privacy domain to choose the appropriate privacy obligations for his shared data. For this purpose, we are based on the existing privacy legislation [10] and standards [5] [9]

Algorithm 1: Privacy_Policy $PPolicy$ **Privacy Attribute Matching** (Access_Request $ARequest$)

$RulesforRequest \leftarrow \varnothing$; $PPolicy \leftarrow \varnothing$
foreach *(output in DeviceOutputs)* **do**
  if *(output==ARequest.requested_data)* **then**
    $RulesforRequest+ = output.hasPrivacyRule$
  end
end
if *(*$RulesforRequest == \varnothing$*)* **then**
  return $PPolicy$
end
foreach *(*$ruleforRequest$ *in* $RulesforRequest$*)* **do**
  if *(*$ruleforRequest.cons$==$NoNeedForConsent$ ***or*** $ARequest.hasConsentResponseDecision$==$Permit$*)* **then**
    foreach *(*$pAtt$ *in* $PATT$*)* **do**
      switch $pAtt$ **do**
        case *Purpose : Disclosure : Operation* **do**
          if *(*$ruleforRequest.pAtt.attribute_level < ARequest.pAtt.attribute_level$*)* **then**
            return $PPolicy$
          end
        end
        case *Retention* **do**
          if *(*$ruleforRequest.pAtt.ret_duration < ARequest.pAtt.ret_duration$*)* **then**
            return $PPolicy$
          end
        end
        case *Condition* **do**
          if *(*$ruleforRequest.pAtt.allowedRole$!= $ARequest.hasInitiator.hasRole$*)* **then**
            return $PPolicy$
          end
        end
      end
    end
    $RulesforPolicy+ = ruleforRequest$
  end
end
$PPolicy$=createPPolicy($output$,$RulesforPolicy$,$ARequest$,$Permit$)
return $PPolicy$
End

in order to propose a set of inference rules that define the privacy obligations related to the data category and privacy rule. For instance, sensitive data are rich in owner-specific habits. That is why, we define Symmetric_Encryption as a privacy obligation of the data category Sensitive_Data. Thus, our reasoner will add this privacy obligation to the privacy policy whenever the shared data's category is Sensitive_Data.

Once the privacy policy is instantiated and the privacy obligations are added, we need to ensure the policy enforcement. To this end, we define a set of inference rules. For instance, the following rule enforces the retention limitation principle by defining the condition that leads to deny a Privacy_Policy. In other words, the hasAccessDecision becomes Deny when the period between the current time and the take_effect_date of the privacy policy is equal to the retention duration.

Privacy_Policy(?ap) $\wedge$ *hasAccessDecision(?ap, Permit)* $\wedge$ *take_effect_date(?ap, ?ted)* $\wedge$ *hasAccessRequest(?ap, ?ar)* $\wedge$ *Retention(?r)* $\wedge$ *hasIntentionPrivacyAttribute(?ar, ?r)* $\wedge$ *retention_duration_per_day(?r, ?rd)* $\wedge$ *temporal:duration(?ret, "now", ?ted, "days")* $\wedge$ *swrlb:equal(?ret, ?rd)* $\longrightarrow$ *hasAccessDecision(?ap, Deny)*

V. Experimentation

In our work, we created the ontology using Protégé [8], which is an open-source ontology editor used to create, modify, delete and query concepts and individuals of our LIoPY. After creating LIoPY, we define a set of inference rules using the SWRL language. The Jena API [13] is used to manipulate our ontology. Jena is a programming toolkit, using the Java programming language. In order to validate LIoPY's consistency and infer new knowledge, we choose the open source reasoner Pellet [11] as it is available with Protégé.

A. Healthcare Case Study

We illustrated our ideas in the healthcare context, but our ontology is agnostic and can be applied in other IoT contexts. Thus, we describe below a motivating healthcare scenario:

Alice, a 40-year old woman who suffers from a heart disease. Preferred to stay at home, she accepted to use a wireless body sensor that will continuously check her health conditions by measuring her heart rate and her position. The sensor collects these data and sends them to the home-gateway through a secure channel. From the medical center, Alice's doctor can remotely monitor her health by receiving Alice's heart rate every few minutes. During the treatment period, the doctor can access the data and add some remarks to Alice's results. In the case of a cardiac problem, the smart device alerts the emergency service by sending Alice's heart rate and position. Then, the hospital dispatches an ambulance to help her. The emergency service can disclose Alice's position to the traffic monitoring service in order to ask for the best route to Alice's location and save valuable time.

In addition, the home-gateway enables Alice to adjust her device settings, including permission and access control. In fact, Alice can add additional people to be notified in case of emergency, such as some of her family members. However, Alice is afraid that one of the authorized people uses her device to monitor her position even in the absence of an emergency.

B. Experimental results

With our experimental stage, we want to first proof the LIoPY feasibility, and second to measure its performance.

In the "Alice LIoPY instance" base, we can find Alice as an individual of the Owner class that owns a Heartrate_Sensor, which is an individual of the sosa:Sensor class. This sensor has two device outputs, which are Alice_Heartrate and Alice_Position. A privacy rule is defined for both device outputs. The first privacy rule, called Privacy_Rule_Heartrate is a rule that grants the permission to remotely monitor Alice's heart rate during 6 months of treatment. The second rule, named Privacy_Rule_Position_Emergency is a rule that grants the permission to collect Alice's position only to the Emergency_Service in the case of a cardiac problem. After launching a set of test sequence batteries, the appropriate privacy policies are derived and one of the obtained derivations is illustrated in Figure 5. This result proves that our reasoning

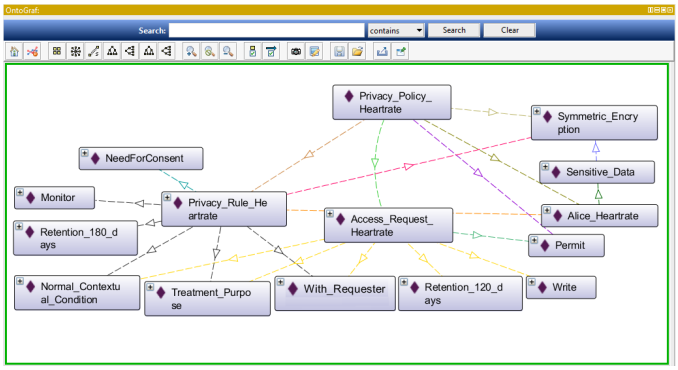


Fig. 5: After Privacy Policy Derivation

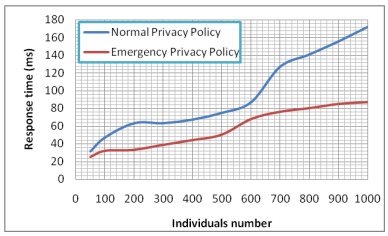


Fig. 6: Response time of Privacy Policy Derivation

process succeed at inferring a privacy policy in case of a match between privacy rules and a request.

After proving the LIoPY feasibility, we conduct some experiments to measure the performance of our solution. To this end, we intend to evaluate if the computing time of reasoning is acceptable by making several tests using a different number of individuals. Hence, we perform an experiment to measure the required time to check and create the privacy policy for two different types of requests: monitor vital sign for normal condition and read position in the emergency case. The number of individuals increases from 50 to 1000. The response time is equal to the consistency checking plus reasoning time.
Figure 6 shows the response time of a privacy policy derivation in normal and emergency contexts. The response time varies from 20 to 180 ms. We observe that our solution can support a large number of individuals within a reasonable reasoning time. For both request types, the difference between having 50 individuals and 1000 individuals is around few milliseconds. This lets us conclude that the increase of individuals does not affect the performance of our solution. Moreover, the linearity property behind these results means that a better computer system setting would obtain a lower reasoning time.

The obtained results demonstrate the LIoPY capability to be instantiated in a real environment for preserving IoT privacy, while overcoming the existing model limitations [1] [4].

VI. Conclusion

IoT emergence presents an opportunity to improve efficiency and quality of life to the users. However, the analysis of the detailed data generated by the IoT resources raises the privacy risks. Semantic modeling is a used solution to give the data owner the control over his data. Moreover, the data owner has not enough experience and expertise in the privacy domain to take advantage of his legal rights. Thus, semantic modeling becomes fundamental to infer the required privacy obligations to preserve privacy. For these reasons, we have proposed a new privacy ontology called LIoPY that aims at defining a common privacy vocabulary combined with standard reasoning technologies based on description logic to address the privacy requirements in the IoT environment. LIoPY is defined over standardized concepts that are extended to protect privacy during the whole lifecycle of the collected data by the IoT resources. LIoPY is experimented in the healthcare context, but it can be applied in other IoT contexts.

Our privacy ontology is developed in an ongoing IoT research project. In our future work, we intend to improve our inference system by proposing new rules ensuring rules conflict detection, such as duplication and contradiction. Besides, we plan to propose a privacy query rewriting algorithm to apply privacy obligations before sharing the device outputs.

References

[1] Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K.: Iot-lite: a lightweight semantic model for the internet of things. In: Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences. pp. 90–97. IEEE (2016)
[2] Evans, D.: The internet of things how the next evolution of the internet is changing everything (april 2011). White Paper by Cisco Internet Business Solutions Group (IBSG) (2012)
[3] Haller, A., Janowicz, K., Cox, S., Le Phuoc, D., Taylor, K., Lefrançois, M., Atkinson, R., García-Castro, R., Lieberman, J., Stadler, C.: Semantic sensor network ontology (w3c recommendation 19 october 2017). https://www.w3.org/TR/vocab-ssn/ (2017)
[4] Hosseinzadeh, S., Virtanen, S., Díaz-Rodríguez, N., Lilius, J.: A semantic security framework and context-aware role-based access control ontology for smart spaces. In: Proceedings of the International Workshop on Semantic Big Data. p. 8. ACM (2016)
[5] International Organization for Standardization: Information technology security techniques privacy framework, ISO/IEC 29100 (2011)
[6] Kotis, K., Katasonov, A.: An ontology for the automated deployment of applications in heterogeneous iot environments. Semantic Web Journal (SWJ) (2012)
[7] Maras, M.H.: Internet of things: security and privacy implications. International Data Privacy Law 5(2), 99 (2015)
[8] Musen, M.A.: The protégé project: a look back and a look forward. AI matters 1(4), 4–12 (2015)
[9] Organisation for Economic Co-operation and Development: Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. OECD (1981)
[10] Regulation, G.D.P.: Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. Official Journal of the European Union (OJ) 59, 1–88 (2016)
[11] Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 51 – 53 (2007)
[12] Smart Grid Interoperability Panel Cyber Security Working Group and others: NISTIR 7628 Guidelines for smart grid cyber security. Privacy and the smart grid 2 (2010)
[13] The Apache Software Foundation: The Apache Jena2 Ontology API. https://jena.apache.org/documentation/ontology/
[14] Wang, S., Hou, Y., Gao, F., Ma, S.: Ontology-based resource description model for internet of things. In: Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016 International Conference on. pp. 105–108. IEEE (2016)