Data Provenance Model for Internet of Things (IoT) Systems

Habeeb Olufowobi[2], Robert Engel[1], Nathalie Baracaldo[1], Luis Angel D. Bathen[1], Samir Tata[1], Heiko Ludwig[1]

[1] Almaden Research Center, IBM Research, San Jose, CA, USA
{engelrob, baracald, bathen, stata, hludwig}@us.ibm.com
[2] Howard University, 2300 Sixth St. NW Washington, DC 20059, USA
habeeb.olufowobi@bison.howard.edu

Abstract. Internet of Things (IoT) systems and applications are increasingly deployed for critical use cases and therefore exhibit an increasing need for dependability. *Data provenance* deals with the recording, management and retrieval of information about the origin and history of data. We propose that the introduction of data provenance concepts into the IoT domain can help create dependable and trustworthy IoT systems by recording the lineage of data from basic sensor readings up to complex derived information created by software agents. In this paper, we present a data provenance model for IoT systems that is geared towards providing a generic mechanism for assuring the correctness and integrity of IoT applications and thereby reinforcing their trustworthiness and dependability for critical use cases.

Keywords: Provenance, IoT

1 Introduction

The Internet of Things (IoT) [7] has received significant attention in industry and in the academic community in recent years. As IoT applications are increasingly deployed for critical use cases, the importance of designing dependable and verifiable IoT systems that can be "trusted" with critical decision-making processes and corresponding automated actions becomes more prevalent.

Consider, for instance, the case of an "intelligent home" equipped with sensors to monitor the health of an elderly individual living in that home. The readings from the sensors are used in conjunction with a real-time analytics engine in order to notify relatives and health professionals when a medical emergency, such as heart attack, is automatically detected by the system. In order to take appropriate action, it is important that the recipients of such a notification of an emergency can trust in the reliability and accuracy of the information reported by the health monitoring system. However, without further *verifiable information* about which sensor data led to the detection of the heart attack it remains unclear on which grounds the system's decision to notify about an emergency is based upon. This lack of transparency impedes the (manual or automated) verification of the correctness of the system's decision. Moreover, if the circumstances that led to the emergency notification (or the lack thereof) are unclear it may not be possible to rule out that the system's decision is an outcome of a malicious attack tampering with the system or a result of otherwise corrupted data, e.g., from a defective sensor.

Data provenance deals with the recording, management and retrieval of information about the origin and history of data [2]. We propose that the introduction of data provenance concepts into the IoT domain can help create dependable and trustworthy IoT systems. This is achieved by recording the lineage of data, starting from the most basic entities of information in the IoT system (e.g., a sensor reading) up to complex derived information (e.g., a notification of an emergency created by a real-time analytics engine and based on various sensor readings). Data provenance is *metadata* and as such requires a data model that describes which specific information is collected.

In this paper, we present a data provenance model for IoT systems. Note that in order to reliably prevent tampering and/or fully verifying information integrity of an IoT system, cryptographic methods such as encryption, hashing, or *blockchaining* need to applied in combination with a data provenance model; however, this is beyond the scope of this paper.

Additional use cases for data provenance in the IoT domain include:

1. *Auditing* and *Digital Forensics*, in which the status of an IoT system or a sequence of events in an IoT system at some time or in some time period of the past needs to be determined (e.g., for investigating the circumstances that were causal for a particular action performed by an IoT application); and
2. *Privacy* and *Data Sovereignty*, in which the origin and lineage of data is instrumental for enabling fine-grained access control for sensitive information collected in IoT applications (e.g., fine-grained access control for health-related data).

2 State of the Art

Data provenance has been a research topic for many years and has been broadly researched in e-science [13], file systems [12], databases [2, 14] and sensor networks [8]. Groth et al. [6] presented a logical architecture for provenance systems identifying key roles (i.e., actors) and their interactions. In an unpublished work, Bauer [1] developed an architectural model for data provenance in the IoT domain. The model defines components for provenance event handling, such as collection, verification, selection and categorization algorithms.

As of today there is no generally accepted form of representing provenance information. Existing approaches for modeling data provenance include the Open Provenance Model (OPM) [10] and the Provenance Data Model (PROV-DM) [11]. OPM provides an ontology for modeling provenance as an annotated graph based on three types of nodes: Artifacts, Processes, and Agents. The edges of these nodes are directed and represent causal dependencies. PROV-DM is a refinement of OPM and aims at covering a broader range of application domains than OPM. PROV-DM models are based on three basic classes: Agents, Activities, and Entities. In addition, PROV-DM provides several predefined relations that can be used in different application domains. Models based on OPM and PROV-DM have been successfully used for capturing provenance in workflow systems (e.g., ProvONE [4], D-PROV [9], P-PLAN [5]). OPM and PROV-DM adopt a largely document-centric view that is centered on the history of actions performed on documents (e.g., for auditing). However, data provenance for dependable IoT systems requires focusing on infrastructures of agents such as sensors, devices, software agents, etc. and the data exchanged between these agents. Moreover, while OPM and PROV-DM provide a rich ontological infrastructure for representing many different types of activities, in the context of dependable IoT systems it is typically sufficient to record only the lineage of data (i.e., its creation and modification).

Other works on data provenance in the IoT domain include the Semantic Sensor Network Ontology (SSN) [3] describing sensors and corresponding observations/values. However, it falls short of modeling other actors in IoT systems, such as devices, software agents, persons or organizations.

3 Data Provenance Model

In this section, we describe a general data provenance model (Section 3.1) as well as preliminary concretization of this model for typical IoT environments (Section 3.2). In Section 3.3 we show an example application of the model.

3.1 General Data Provenance Model

We define a *data point* (dp) as a uniquely identifiable and addressable piece of data (i.e., a value) in the context of an IoT system. Examples for dps in the context of IoT systems include sensor readings such as discretized audio/video streams, complex analytics results derived from sensor readings, actuator commands, etc. A dp distinguishes itself specifically from other data flowing in the system (e.g., bulk sensor readings that are of no further interest, ephemeral intermediary analytics results, etc.) in that it is addressable, i.e., it has an ID that is unique in the context of the IoT system. A dp may be based on other dps that have contributed to its creation or modification. We refer to these related dps as *input data points* (input dps). We introduce the *addr(dp)* and *inputs(dp)* functions providing the address/ID of a dp and the set of input dps, respectively.

While our proposed model is independent of a specific execution model of IoT applications (e.g., event-driven, workflow-driven, etc.), the execution logic of an IoT application is responsible for defining *when (if)* provenance information needs to be collected for a particular dp. We refer to the specific state of an IoT system in which the collection of provenance data for a particular dp is deemed necessary by the execution logic as a *provenance event.*

We define *context* of a dp as information about state of an IoT system that is of interest for provenance when a corresponding provenance event occurs; we write $context(dp)$ to denote a corresponding function. Context is the “core” of the provenance information for a dp and its specific contents may vary for different IoT applications. For instance, context may include information about agents involved in the computation of the dp, time and location information, execution context (e.g., triggering events), etc. While the specific contents of context are not defined in this general data provenance model, we show a possible concretization of context for IoT applications in Section 3.2.

We define a *provenance record* as a tuple associating the address of a dp with the set of provenance records of its input dps and the specific context of the corresponding provenance event. We introduce the *provenance function* $prov(dp)$ providing the provenance record for some dp as

$prov : dp \mapsto \langle addr(dp), \{prov(idp) | idp \in inputs(dp)\}, context(dp)\rangle$.

Note that this definition of provenance allows for the description of both *creation* and *modification* of dps. In the latter case, the set of input dps contains the provenance record of the dp before modification, i.e.,

$prov(dp') = \langle addr(dp'), \{prov(dp), ...\}, context(dp').\rangle$.

3.2 Context

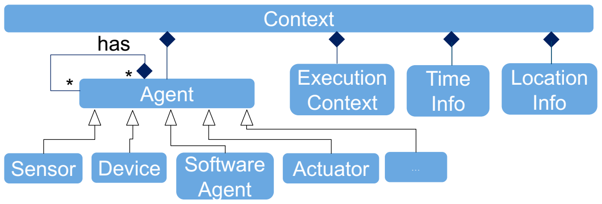

Fig. 1. Class diagram of a possible model for *context*

Specific context for provenance may vary for different IoT applications. We propose an early-stage data model for context in typical IoT environments comprising the concepts of *Agents*, *Execution Context*, as well as *Time* and *Location* information (cf. Figure 1).

An *Agent* is an entity that creates and/or modifies data points (e.g., sensor, software agent, device, person, organization, etc.). It is recursively defined such that an agent may contain other agents (e.g., a device containing several sensors). This recursion allows for defining agents in a hierarchy, and may be used as fine-grained as required. For instance, an agent hierarchy may span from the concept of a particular function in a software library running over a virtualization container on a particular device to a particular IoT network.

Execution Context provides information related to the provenance event at hand, such as events or dps that triggered the creation/modification of the dp; this information may be specifically required for use cases related to auditing and digital forensics. *Time* and *Location* information may be added to the provenance information for a particular dp.

3.3 Example

Consider, for instance, an IoT application responsible for computing the average temperature in a room over some time period. There are three readings from a temperature sensor: $dp_1 = 77F, dp_2 = 55F, dp_3 = 63F$. According to the definition in Section 3.1, their provenance records are given as follows:

$prov(dp_1) = \langle addr(dp_1), \emptyset, \langle agent = sensor1@raspPi1, time = 5am, ...\rangle\rangle$
$prov(dp_2) = \langle addr(dp_2), \emptyset, \langle agent = sensor1@raspPi1, time = 6am, ...\rangle\rangle$
$prov(dp_3) = \langle addr(dp_3), \emptyset, \langle agent = sensor1@raspPi1, time = 7am, ...\rangle\rangle$

A fourth datapoint dp_4 is created by a software agent responsible for calculating the average temperature, i.e., $dp_4 = (dp_1 + dp_2 + dp_3)/3 = 65F$. Hence, the provenance record of dp_4 is given by:

$prov(dp_4) = \langle addr(dp_4), \{prov(dp_1), prov(dp_2), prov(dp_3)\},$
$\langle agent = averager@cloudvm1, time = 8am, ...\rangle\rangle$

Thus far, the provenance events for dp_1 to dp_4 were related to the *creation* of new datapoints in the IoT application. In other cases dps may be *modified* instead, such as a software agent converting units of the calculated average temperature: $dp'_4 = 17.78C$. The provenance record for this updated dp is given as follows:

$prov(dp'_4) = \langle addr(dp'_4), prov(dp_4),$
$agent = converter@cloudvm1, time = 9am, ...\rangle\rangle$

4 Implementation

IoT systems are composed of sensors, actuators, devices, and gateways that are possibly connected to a Cloud. Managing provenance in such a system requires the provisioning of mechanisms that collect, store and query provenance data. To do so, we are currently implementing provenance mechanisms based on the MQTT[3] broker which should be deployed in any IoT component where provenance data should be managed. Those IoT components are clients and providers of provenance data. They are provided with a simple, lightweight, publish/subscribe messaging protocol and system and local provenance data stores. Each IoT component can subscribe for any provenance data topic coming from any other IoT component. IoT components are also responsible of sending provenance data for all registered (and authorized) IoT components.

As mentioned before, our proposed model for data provenance is independent of a specific execution model of IoT applications. The execution logic is responsible for defining when (if) provenance information needs to be collected. To support that, we are implementing declarative provenance management policies to describe what provenance data should be collected. Moreover, such policies control how the provenance data is disseminated and stored in the IoT system.

[3] http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

5 Conclusion

In this paper, we presented a data provenance model for IoT systems. The model describes the context of the creation or modification of datapoints, including information about involved agents, execution context, time, and location information. Moreover, the model captures dependency relationships between datapoints and is independent of execution models of IoT applications.

By ensuring the traceability of the origin of data as well as capturing metadata about the processes that result in derived information - in future work combined with cryptographic methods for verifying the integrity of provenance metadata - we aim at providing a generic mechanism for assuring the correctness and integrity of IoT applications and thereby reinforcing their trustworthiness and dependability for critical use cases.

References

1. Bauer, S., Schreckling, D.: Data provenance in the internet of things. In: EU Project COMPOSE, Conference Seminar (2013)
2. Buneman, P., Khanna, S., Wang-Chiew, T.: Why and where: A characterization of data provenance. In: Intl. Conf. on Database Theory. pp. 316–330. Springer (2001)
3. Compton, M.e.a.: The ssn ontology of the w3c semantic sensor network incubator group. Web Semantics: Science, Services and Agents on the World Wide Web 17, 25–32 (2012)
4. Cuevas-Vicenttín, V., et al.: Provone: A prov extension data model for scientific workflow provenance (2015)
5. Garrijo, D., Gil, Y.: P-plan ontology (2012)
6. Groth, P., Jiang, S., Miles, S., Munroe, S., Tan, V., Tsasakou, S., Moreau, L.: An architecture for provenance systems. Tech. rep. (2006)
7. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (iot): A vision, architectural elements, and future directions. Future Generation Computer Systems 29(7), 1645 – 1660 (2013)
8. Lim, H.S., Moon, Y.S., Bertino, E.: Provenance-based trustworthiness assessment in sensor networks. In: Seventh International Workshop on Data Management for Sensor Networks. pp. 2–7. DMSN '10, ACM, New York, NY, USA (2010)
9. Missier, P., Dey, S., Belhajjame, K., Cuevas-Vicenttín, V., Ludäscher, B.: D-prov: Extending the prov provenance model with workflow structure. In: 5th USENIX Workshop on the Theory and Practice of Provenance (TaPP 13) (2013)
10. Moreau, L., et al.: The open provenance model core specification (v1. 1). Future generation computer systems 27(6), 743–756 (2011)
11. Moreau, L., et al.: Prov-dm: The prov data model. w3c recommendation (2013)
12. Muniswamy-Reddy, K.K., Holland, D.A., Braun, U., Seltzer, M.I.: Provenance-aware storage systems. In: USENIX Annual Technical Conf. pp. 43–56 (2006)
13. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Rec. 34(3), 31–36 (Sep 2005)
14. Tan, W.C., et al.: Provenance in databases: Past, current, and future. IEEE Data Eng. Bull. 30(4), 3–12 (2007)