Identity Management in Platforms Offering IoT as a Service

Juan D. Parra Rodriguez(✉), Daniel Schreckling, and Joachim Posegga

Institute of IT-Security and Security Law, University of Passau,
Innstraße 43, Passau, Germany
{dp,ds,jp}@sec.uni-passau.de

Abstract. We describe a generic attribute-based identity management system. It aims to support the large variety of security requirements induced by applications for the IoT. Hence, we discuss various management options for system entities. We show how attribute assurance can be used to reliably define attributes within groups of identities. Apart from enabling personalized identity and policy enforcement schemes, this provides a feasible trade-off between the flexibility and scalability needs and the policy definition and enforcement requirements in the IoT. We provide a proof-of-concept implementation of our framework.

Keywords: Identity management · Internet of Things · Platform as a service · Attribute based access control · Federated identity management

1 Introduction

There is a growing need for an Internet of Things (IoT) runtime ecosystem in which services using actuators and data from sensors can coexist. Such an ecosystem would not only expose functionalities as reusable services. It could also provide usage control; ensuring that data processing is compliant with the security requirements of individual device owners. These inherently diverse requirements, ask for fine-grained and expressive security policy frameworks. Such security frameworks, in turn, require a flexible identity management (IDM) system for *things*, services processing IoT-data, owner or users of *things*, and other related entities. Thus, we propose an infrastructure to identify, authenticate, and manage the security principals in an IoT platform. The three main contributions of this paper can be summarized as follows:

- Identification of challenges faced when users define their own IDM scheme in a platform offering IoT as a service.
- Analysis of possible approaches to enable users to share identity information with each other, while comparing the computational complexity induced in the policy decision process.
- Providing mechanisms designed to ensure the reliability of identity information and enabling users to share such information, without a significant increase of computational complexity in the policy evaluation process.

R. Giaffreda et al. (Eds.): IoT360 2014, Part I, LNICST 150, pp. 281–288, 2015.
DOI: 10.1007/978-3-319-19656-5_40

2 Challenges

2.1 Support for Access and Usage Control

IoT platforms have to manage and process data of myriads of devices used for different applications. Controlling access to such devices using mandatory access control (MAC) is infeasible. In contrast, discretionary access control (DAC) needs to offer the flexibility required for different application for independent users. While role based access control (RBAC) may efficiently combine both worlds, it strictly requires a unified hierarchy of roles.

Fortunately, attribute based access control (ABAC) [1–3] is able to merge the abilities of MAC, DAC, and RBAC [4] without imperatively enforcing a centralized role management. It enables technologies which grant access based on attributes assigned to subjects, objects, or the environment. Flexible and domain specific security policies can be defined easily. With an IDM supporting attributes it is also possible to implement sophisticated systems which support fine-grained policy decisions such as $UCON_{ABC}$ model [5]. This is beneficial for the processing of security sensitive data in IoT platforms as coarse grained policies based on system principals or resource levels are insufficient and inflexible. Instead, attributes enable domain and application specific control mechanisms. Policies and their enforcement can be adapted to specific needs by considering individually defined and trusted attributes.

As a consequence, the IDM presented in this contribution addresses the specific needs and security enforcement technologies feasible for IoT systems. Thus, a reliable attribute-based principal management that enforces the correct and authorized declaration and definition of attributes becomes inevitable.

2.2 Attribute Assurance

Once users can define their own identity schemes in the platform, they start to resemble small identity 'silos' with their own self-defined attributes. As a result, enabling users to interact and exchange identity information becomes critical. To exemplify the challenges to be tackled, we consider the following scenario.

Two companies provide sensor data through an IoT platform: *SensIoT*, and *fakeIoT*. Further, assume each company creates an attribute called "produced by" containing the name of the company. Most likely, sensor containing the attribute "produced by" with value *SensIoT* will be used more. However, unless there is a mechanism enforcing that only devices belonging to *SensIoT* can assign the value *SensIoT* to the attribute "produced by", *fakeIoT* could start faking attribute values to fool customers. Even though this example could be solved by including a simple customized verification, the question is, how to enable companies, or groups of users, to define attributes which require approval before those attributes are considered reliable. We call this process *attribute assurance*, given its affinity to a more process driven aspect of Federated Identity Management (FIDM), called identity assurance [6,7].

2.3 Sharing Attributes with Other Users

Users should be able to decide whether they consider attribute values approved by other users, or groups of users, to be sufficiently reliable for supporting policy decisions. This is a parallel challenge to managing trust in FIDM [6] with an additional pitfall: the inclusion of users' preferences should not increase the complexity of policy decisions. We discuss two possibilities to manage trusted attributes and analyse their computational complexity. For this purpose, let n be the number of attribute values for a device, service, or user that require approval according to a policy[1] and let the cost of verifying the approval status for a given attribute be in *O(1)*.

The first possibility is to use a **global and centralized repository of attribute values** and to authorize selected users to globally approve attributes. In this case policy evaluation could be in *O(n)*. Only the verification of an approval flag would be required for each attribute value. Policy evaluation becomes trivial as an approved attribute is automatically trusted system-wide. However, authorized users won't be elected freely, declining the possibility to benefit from approval mechanisms to most of the users in the IoT platform; also, the attribute approvals become unmanageable for a limited number of authorized users.

A completely opposite approach is to allow users to set and approve attribute values. In such an **ad-hoc sharing scheme for attributes** the most important question is whether an attribute value has been approved by a trusted user. This notion can be implemented by adding credibility to attributes [8]. Users specify a list of trusted users for each attribute used in a policy. Let k be the number of users in the system. K is both, the upper bound of attribute approvals for a given attribute value, as well as the upper bound of trusted users chosen for each one of the n attributes. In this case, the computational complexity of the policy evaluation is in $O(n \cdot k^2)$. For each approval, it must be verified whether a trusted user approved the value. Although flexible, this approach induces high computational complexity during evaluation. Hence, we propose a trade-off between flexibility required for IoT environments and the complexity imposed on policy evaluation.

3 Identity Management Scheme

Identities are simply a computer's representation of an *entity* [9]. They are required when their corresponding entity is either providing some functionality or involved in some access/usage control decision. Thus, in our IoT platform, we consider users, device representation (sensors, actuators), and services. Including these entities beyond users allows us to control access/usage on data generated by devices, services or the devices, and services themselves.

[1] Filtering which attribute values from the user, service or device correspond to the attributes referenced by the policy is disregarded (same algorithm for every case).

3.1 Entity Management

Users can create their own groups (e.g., organizations, family, friends). To have significance for security, groups must implement access control mechanisms. This is achieved trough mutual agreement on group memberships.

A group membership is defined as a tuple *(u,r,g)* where user u has role r in group g. Further, before memberships are considered effective by the IDM system, it has to be approved by two parties: the administrator of g, and u. In this way users cannot be misplaced in a group, and groups only contain approved users. Due to this property, groups are the cornerstone for attribute assurance and for the sharing of attribute values among users (Sects. 3.3 and 3.4).

3.2 System Structure

The IDM system contains eight modules shown in Fig. 1. Every API call to the prototype comes from the bottom of the picture; consequently, it must go through the authentication and authorization module before it reaches any other module. Afterwards, if the request is related to the creation or deletion of a user, group, or an entity, it will go directly to the appropriate registry. Note that users are both in the user and entity registry for readability reasons, and also because they could be handled by a third party, such as an user authentication server.

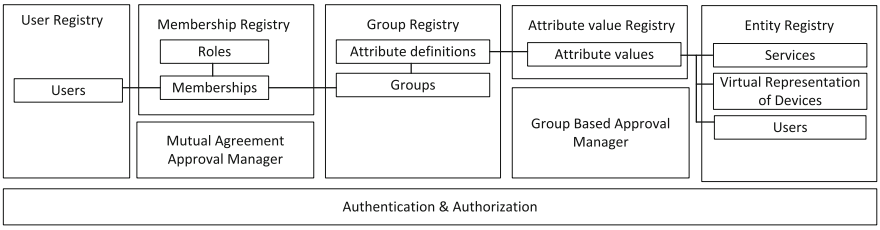

Fig. 1. Internal identity management architecture

If there is a request for the creation, deletion or approval of a user membership, it must go through the *Mutual Agreement Membership Manager*, which will take care that memberships are updated only by the right users. Similarly, if the request concerns the approval of an attribute, it will go through the *Group Based Approval Manager*, which judges whether the user trying to approve an attribute value has the right permissions in the correct group.

3.3 Attribute Definition and Assurance

Figure 2 shows how attribute definitions, attribute values, and groups are related to each other. Attributes are defined in a per-group basis. For attribute assurance purposes, attribute values are only valid after they have been approved

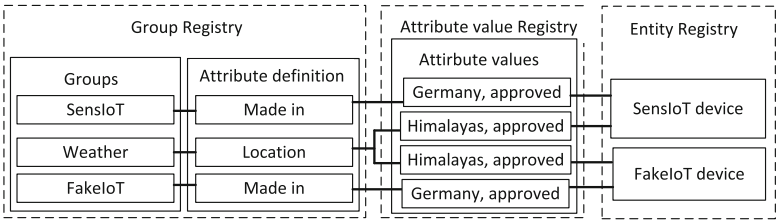

Fig. 2. Attribute definition

by administrators of the group where the attribute definition is registered in. Figure 2 shows two groups (SensIoT and FakeIoT) that have an attribute called "Made in", and also two devices with approved attribute values for the "Made in" attribute definitions in each group. Also, both devices have values pointing to the same attribute definition (approved by the same authority) for the "Location" attribute, previously defined inside the weather group. This implies an approval from an administrator of Weather group for "Himalayas" (both devices).

3.4 Sharing Attributes with Other Users

Given the properties described in the previous section, groups are employed as a mechanism to let users express which users (in a group) are considered trusted. Here is the advantage in terms of policy evaluation: when users refer to the attribute definition directly in their policies (e.g. "Made in" attribute definition inside SensIoT), the complexity[2] of the policy evaluation algorithm is *O(n)* since for every attribute value, only the approval state of the attribute value needs verification. Most interestingly, attribute assurance is not lost since only administrators from the group can approve the attribute value. Further, the proposed schema is still flexible enough to support an ad-hoc approval where users provide just the name of the attribute, the expected value, and the trusted groups for approval in their policies; nevertheless, this is not encouraged due to computational complexity (see Sect. 2.3).

3.5 Access Control Integration

We briefly illustrate the power of our IDM for policy enforcement. Assume a variety of sensors deployed in an alpine area. They have a virtual representation in our platform and can be used by different services. Further, suppose independent service providers use these sensors to report on slope conditions, generate warnings for mountaineers, or shutdown cable cars. A slope condition service simply uses attributes to determine whether a sensor is on a specific slope. Such attributes may change over time and confuse users but they are not

[2] n is number of attribute values for the entity, such as the SensIoT device.

critical. However, safety relevant services need to make sure only sensors with assured attributes, such as the height or location, are used. This particularly holds for services casting warnings or taking actions. For this purpose, service providers can define policies described by simple predicates validating attribute and checking their values. In this way ABAC policies can be defined also supporting the basic principles of $UCON_{ABC}$ policies. Thus, instead of accumulating the potentially changing set of IDs of entities allowed to access another entity and assigning them some property, our IDM supports direct filtering based on attributes. This simplifies the task to implement domain specific access control, avoids administrative overhead, and prevents burdensome reimplementaitons.

As we offer IoT services hosted in one platform we also support decentralized evaluation of policies at the entity level. During entity deployment local PDPs extract required information from the registries tagged to the respective entities. In this way, access control decision on services, for example, boil down to simple and localized value comparisons.

4 Related Work

Contributions using attributes in IDM systems are manifold and were inspired by the first designs of ABAC [10]. Based on these models implementations for ABAC have been proposed. The prominent contribution by Bonatti and Samarati [11] introduced a framework to logically specify and reason about service access and disclosure constraints in open environments. While this contribution recognizes the relevance of ABAC for $UCON$ and the flexibility to describe entities of a system with attributes, it neither focuses on their management nor does it consider their ad-hoc definitions. Similar holds for the modelling of ABAC in a logic-based framework [1] where policies defined on attributes are logic programs with recursion allowing for powerful and efficient evaluation.

Yuan and Tong apply ABAC to the web service domain [2]. An authorization and policy definition architecture is described but it remains unclear how attributes could be defined in a flexible manner. ABMAC [12], an extension of the ABAC model, faces the same problem. Comparable to our IDM, ABMAC also aims towards the support of multiple heterogeneous policies. Yet, it does not consider attribute assurance as presented in this contribution.

Another popular approach is the use of attribute certificiates [13–17]. They bind entity attributes and their values to a domain. Apart from the propblem to setup up a commonly accepted PKI, these solutions induce impractical overheads in terms of management, during policy evaluation and challenge scalability.

In contrast, Cabarcos et al. introduce a dynamic scheme for federations among IDM systems [18]. They define a P2P reputation system with dynamic trust lists. If entities out of this list provide attribute values, reputation values can be queried to decide on the reliability of the information. Reputation is defined by domain administrators. Although this system enables dynamic and gradual trust, users cannot individually administer their policy domain.

A similar approach is taken in the Liberty framework, a federated IDM system. It uses the Identity Web Service Framework (ID-WSF) [19] to store

principal attributes which can also be used for policy decisions. However, the attribute declaration is strongly centralized and only security providers can define new attributes and does not allow to select specifically trusted attributes or authorities.

5 Conclusion

We have designed and implemented a prototype[3] for a generic attribute-based IDM for a platform offering IoT as a service. In such system, every user can define his own attribute scheme, and use mechanisms to ensure reliability of attribute values to other users in the platform. Furthermore, users of the platform can choose trusted groups to approve attribute values before they are used by the PDP. Additionally, the proposed solution offers a good trade-off between flexibility and complexity increase for the policy evaluation process.

Acknowledgements. The research leading to these results has received funding from the European Union's FP7 project COMPOSE, under grant agreement 317862.

References

1. Wang, L., Wijesekera, D., Jajodia, S.: A logic-based framework for attribute based access control. In: Proceedings of the ACM Workshop on Formal Methods in Security Engineering, FMSE 2004, pp. 45–55. ACM, New York (2004)
2. Yuan, E., Tong, J.: Attributed based access control (ABAC) for web services. In: Proceedings of the IEEE International Conference on Web Services, pp. 561–569, July 2005. doi:10.1109/ICWS.2005.25
3. Hu, V.C., Scarfone, K., Kuhn, R., Sandlin, K.: Guide to attribute based access control (ABAC) definition and considerations. Technical report, Nation Institute for Standards and Technologies, January 2014
4. Jin, X., Krishnan, R., Sandhu, R.: A unified attribute-based access control model covering DAC, MAC and RBAC. In: Cuppens-Boulahia, N., Cuppens, F., Garcia-Alfaro, J. (eds.) DBSec 2012. LNCS, vol. 7371, pp. 41–55. Springer, Heidelberg (2012)
5. Park, J., Sandhu, R.: The $UCON_{ABC}$ usage control model. ACM Trans. Inf. Syst. Secur. **7**(1), 128–174 (2004)
6. Jensen, J.: Federated identity management challenges. In: Seventh International Conference on Availability, Reliability and Security, pp. 230–235. IEEE, August 2012
7. Beres, Y., Baldwin, A., Mont, M.C., Shiu, S.: On identity assurance in the presence of federated identity management systems. In: Proceedings of the ACM Workshop on Digital Identity Management, DIM 2007, pp. 27–35. ACM, New York (2007)
8. Thomas, I., Meinel, C.: Enhancing claim-based identity management by adding a credibility level to the notion of claims. In: 2013 IEEE International Conference on Services Computing, pp. 243–250 (2009)

[3] https://github.com/nopbyte/compose-idm.

9. Bishop, M.A.: The Art and Science of Computer Security. Addison-Wesley Longman Publishing Co., Inc., Boston (2002)
10. Johnston, W., Mudumbai, S., Thompson, M.: Authorization and attribute certificates for widely distributed access control. In: Proceedings of the 7th Workshop on Enabling Technologies, pp. 340–345. IEEE Computer Society, Washington, D.C. (1998)
11. Bonatti, P.A., Samarati, P.: A uniform framework for regulating service access and information release on the web. J. Comput. Secur. **10**(3), 241–271 (2002)
12. Lang, B., Foster, I., Siebenlist, F., Ananthakrishnan, R., Freeman, T.: A flexible attribute based access control method for grid computing. J. Grid Comput. **7**(2), 169–180 (2009)
13. Thompson, M.R., Essiari, A., Mudumbai, S.: Certificate-based authorization policy in a PKI environment. ACM Trans. Inf. Syst. Secur. **6**(4), 566–588 (2003)
14. Chadwick, D.W., Otenko, A.: The PERMIS X.509 role based privilege management infrastructure. In: Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, SACMAT 2002, pp. 135–140. ACM, New York (2002)
15. Alfieri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Frohner, A., Gianoli, A., Lõrentey, K., Spataro, F.: VOMS, an authorization system for virtual organizations. In: Fernández Rivera, F., Bubak, M., Gómez Tato, A., Doallo, R. (eds.) Across Grids 2003. LNCS, vol. 2970, pp. 33–40. Springer, Heidelberg (2004)
16. Guo, S., Lai, X.: An access control approach of multi security domain for web service. Procedia Eng. **15**, 3376–3382 (2011)
17. Cha, B.R., Seo, J.H., Kim, J.W.: Design of attribute-based access control in cloud computing environment. In: Kim, K.J., Ahn, S.J. (eds.) Proceedings of the International Conference on IT Convergence and Security. Lecture Notes in Electrical Engineering, vol. 120, pp. 41–50. Springer, Netherlands (2012)
18. Arias Cabarcos, P., Almenárez, F., Gómez Mármol, F., Marín, A.: To federate or not to federate: a reputation-based mechanism to dynamize cooperation in identity management. Wireless Pers. Commun. **75**(3), 1769–1786 (2014)
19. Tourzan, J., Koga, Y. (eds.): Liberty ID-WSF web services framework overview (Version 2.0). Technical report, Liberty Alliance Project (2006)