Managing Access Control for Things: a Capability Based Approach

D. Rotondi
TXT e-solutions S.p.A.
c/o Tecnopolis N.O.
Strada Prov. Casamassima Km 3
70010 Valenzano (BA), IT
+39-02 25771 782
domenico.rotondi@txtgroup.com

S. Piccione
TXT e-solutions S.p.A.
c/o Tecnopolis N.O.
Strada Prov. Casamassima Km 3
70010 Valenzano (BA), IT
+39-02 25771 757
salvatore.piccione@network.txtgroup.com

ABSTRACT
Traditional and widely used access control mechanisms have been proved to be not able to effectively support the dynamicity and scaling needs of IoT contexts. Furthermore, as more end-users start using smart devices (e.g. smart phones, smart home appliances, etc.) the need to have more understandable and easy to use access control mechanisms increases. In this paper we present a capability based access control system, which is being developed in a EU project harnessing IoT technologies in industrial and automation environments, showing that it can better address IoT needs and can be more easily applied to end users-centric scenarios like smart houses and e-Health.

Categories and Subject Descriptors
D.4.6 [**Operating systems**]: Security and Protection – *Access controls, authentication, information flow controls.*

K.6.5 [**Management of computing and information systems**]: Security and Protection – *Authentication, unauthorized access.*

H.1.2 [**Models and Principles**]: User/Machine Systems – *Human factors.*

General Terms
Management, **Reliability**, **Security**, **Human Factors**.

Keywords
Authorization, Access Control, Capability Based Access Control, Rights Delegation, Rights Revocation, Internet of Things.

1. INTRODUCTION
The Internet of Things (IoT) presents novel security challenges that require new solutions or a substantial revision of existing ones. Access control is one of these challenges as we highlight in the following. This paper provides a description of the Capability Based Access Control (CapBAC) system being developed in the EU FP7 IoT@Work project (http://iot-at-work.eu) to address specific issues in an IoT world. In particular, The IoT@Work project focuses on harnessing IoT technologies in industrial and automation environments. In the following sections we provide a quick analysis of the IoT access control issues and a survey of our CapBAC approach in the light of issues more specifically related to consumer' devices (e.g. smart phones, tablets) and scenarios (e.g. domotics, e-Health) for which, we think, CapBAC is more effective and usable as compared to traditional access control approaches. The paper is organized as follows: *Section 2* provides an overview of IoT access control issues, in general, and specific issues related to the indicated consumer's devices and scenarios; *Section 3* provides a quick survey of capability based security and related work; *Section 4* provides an overview of our CapBAC system, of its features and functional elements, as well as some specific aspects related to the indicated scenarios; *Section 5*, finally, reports the CapBAC system current status and future work.

7th International Conference on Body Area Networks (BODYNETS 2012), Workshop on Security Tools and Techniques for Internet of Things (SeTTIT 2012), September 24–26, 2012, Oslo, Norway

2. IoT ACCESS CONTROL ISSUES
The Internet of Things has not only to master a wider heterogeneity of connected systems, communication technologies and resource constraints (for processing, storage and communication), but has also to face issues related to the potential unbounded number of interacting subjects (devices, applications, humans), as well as a substantial difference in the interaction patterns ([1], [2]), which evolve from a predominance of more planned and long-lived to short-lived, often casual or spontaneous ones. Additionally, IoT envisages an enhanced relevance of the context awareness ([3]), higher needs to support the orchestration and integration of different services, as, for example, envisaged by the *DiY* (*Do it Yourself*) sociocultural practice ([4]) and scalability, manageability and usability ([5]).

As far as access control is concerned IoT requires solutions that at least:

- are able to face the IoT scalability challenge;
- are easy to manage;
- can be even deployed on simple devices (e.g.: minimize resources requirements for storage, processing, communication, etc.);
- are secure and flexible;
- support advanced features (e.g.: access rights delegation, auditability, etc.);
- can provide an easy to use interface suitable for consumers and devices needs.

The next paragraphs will provide a concise analysis of traditional access control mechanisms and of some of the requirements listed above.

2.1 Access Control Systems
The objective of any access control system is to provide features to control who (entities that are normally called *subjects*) can do what (actions normally called *operations* or *rights*) on which resources (called *objects*). Therefore an access control system can be modeled, in general, as a set of objects O, which represents the

DOI 10.4108/icst.bodynets.2012.250234

resources whose access has to be controlled, a set of subjects S, which collects the actors (e.g. people, applications, etc.) that can exercise actions on objects in O, a set of rights R, and, finally a set of rules ρ connecting objects, subjects and rights:

$$\rho_i(\sigma, \omega, r) \; where \; \sigma \in S, \; \omega \in O, \; r \in R$$

The rules can be conveniently arranged in the so called *Access Control Matrix* ([6]) a bi-dimensional matrix $M(\sigma, \omega)$that, for each combination of (σ, ω), specifies the *rights subject* σ as on *object* ω.

The most common form of access control is based on access control lists (ACLs) that are being provided by the most widely used operating systems. ACLs become very complex to manage when the number of subjects or resources increases, or in situations in which subjects often change their rights.

To overcome the burden of basic ACLs systems, *Role Based Access Control* (*RBAC*) systems ([7], [8]) were designed; they add an intermediate layer assigning *rights* to roles instead of granting them directly to *subjects*, and then assign roles to *subjects*. This approach can heavily reduce the effort required to manage the access control rules, even if it can still require a lot of effort for example due to roles explosion when the number of resources grows or when the access control system covers a number of administrative domains (e.g. web sites arranged in a *circle of trust* providing a *Single Sign On* service [9]).

Attribute Based Access Control (*ABAC*) systems, well exemplified by the XACML standard ([10]), try to solve the issues of *RBAC* systems making possible to directly use properties associated to subjects (e.g.: age, location, position in an organization, etc.), as well as resources and environmental properties, to specify access rules. *ABAC* systems still require that attributes be defined consistently within a domain or across different domains; additionally, *ABAC* systems require specialized services (e.g. service able to provide subject's attributes) and normally require more processing power and storage capabilities.

All the above systems make hard to enforce the least privilege principle ([11]) (also known as *Principle of Least Authority*) that, as originally defined by Saltzer ([12]), states "*every program and every privileged user of the system should operate using the least amount of privilege necessary to complete the job*". Furthermore, in widely open contexts like SOA and Grid Computing the above access control systems present serious scalability issues and envisage increasing management effort ([28], [13], [14]). Finally they do not provide flexible and easy to use rights delegation features.

2.2 Access Control Systems and Usability

With the widespread usage of ICT in everyday life, the usability of devices (e.g. smart phone, tablets), applications (e.g. iPad/Android apps) and services is becoming a key, distinguishing element. As Adams and Sasse stated ([15]) "*… security mechanisms are designed, implemented, applied and breached by people, human factors should be considered in their design*" and this has to be applied to access control features also, especially when these features move from enterprise's systems and services to consumer's everyday devices and environments. The usability of access control has so far received limited attention ([16]) in the research community, even if in these years some interesting analysis have been performed especially for issues related to home systems ([17], [18]).

For consumer's devices and scenarios, like domotics and e-Health, RBAC/ABAC systems present relevant usability issues. Indeed roles cannot actually be defined in a consistent and general way being each user (almost) autonomous and their meaning and usefulness difficult to be caught. Furthermore these contexts do not usually have enough computing resources to deploy the supporting services RBAC/ABAC systems require (e.g.: access rules repository, policy decision points). Additionally RBAC/ABAC systems require the availability of authentication and identity management features, further increasing the usability barriers for consumers and citizens and resource requirements. Finally these access systems are not able to face the dynamicity present in the everyday life (for example to easily and quickly manage temporary delegation of rights for home appliances when the householder goes on holidays).

3. CAPABILITY BASED SECURITY

Capability based security is a security model in which "*… a capability (known in some systems as a key) is a communicable, unforgeable token of authority. It refers to a value that references an object along with an associated set of access rights*" (see http://en.wikipedia.org/wiki/Capability-based_security).

As depicted in Figure 1, while in a traditional access control system it is the service provider that has to check if the user is, directly or indirectly (for example via a role owned by the user), entitled to perform the requested operation on the requested resource, in a capability based system the user has to provide his/her/its authorization capability (and demonstrate he/she/it owns it) to the service provider.

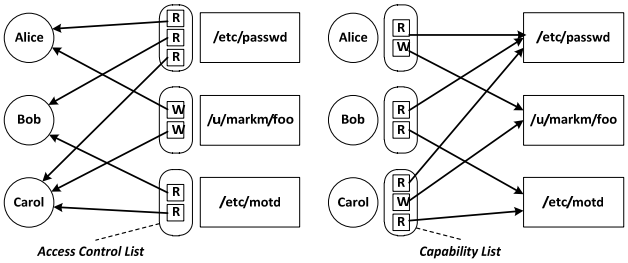

Figure 1. ACL vs Capability-based authorization models.

Thus in a capability based system the service provider (e.g. your smart phone or your home appliance) does not have to manage sets of access rules/policies or be able to support authentication protocols and trust relationships: it is only required to be able to check the access capability correctness, the proof of ownership the requester has provided and the compliance of the access capability with the requested operation on the requested object (of course the service provider has to be able to trust the issuer of the capability. This in many situations simply means trusting the issuer's digital signature).

3.1 Related Work

Capability based security is not a new concept ([19], [20], [21]) and it has been used to devise the *SPKI Certificate Theory* standardized by the RFC2693 [22] document. SPKI defines a sort of PKI focused on an authorization mechanism based on the definition and exchange of a sort of authorization certificate. X.509 has included since 1997 Attribute Certificates ([23], [24]) as a mean to specify subject's information (e.g. group membership, role, security clearance) useful for authorization management. In recent years capability based security models has

been used to address usability issues (see for example the XEROX *Casca* application [25] or the XPOLA access control system [14]) or rights delegation in grid or service oriented systems ([14], [26]). Skinner [13] has proposed a capability based authorization approach to address the dynamicity and scalability issues of *Digital Ecosystem* environments; while Jun Li [27] and Karp ([28], [29], [30]) suggest to use capability based models to address similar issues as well as to effectively address rights delegation. Factor and his IBM colleagues [32] have proposed a capability based access control mechanism for storage area networks to address issues of storage access in Cloud Computing.

A more detailed analysis and comparison of capability based security features and issues against traditional approaches have been provided by Miller and his colleagues in [21].

3.2 Advantages of Capability Based Security

As highlighted above IoT is more demanding in terms of scalability and manageability [5], as well as envisages a wider range of communication and processing capability of the involved entities (smart sensors, smart actuators, smart phone, etc.) and the presence of more casual and spontaneous interaction patterns. These issues directly impacts access control management that can become a nightmare in IoT if not addressed with new approaches.

Indeed, RBAC and ABAC "*... have been found to be inflexible, don't scale well, and are difficult to use and to upgrade*" [28]. Furthermore these systems have substantial management overhead, security issues (e.g. *confused deputy* problem [33], rights revocation), and complex arrangements to support delegation and transitivity, as well as for managing access policies and assure policy compliance ([28], [30]). Capability based access control and rights delegation systems more easily can support the following features:

- the *Principle of Least Authority* (*PoLA*): a capability grants specific set of rights on an *object* to a *subject*. *PoLA* is therefore the default;
- a more fine-grained access control because each *object* can have its how *rights*;
- less security issues (e.g. no *confused deputy* problem);
- can externalize and distribute the management of the authorization process among many *subjects* thanks to the easy support of delegation mechanisms;
- relieves the management of issues related to complexity and dynamics of subject's identities.

Additionally, identity management and authentication issues are less relevant in capability based systems providing huge advantages in cross-domain contexts or in consumers or citizens scenarios.

The main reasons behind our choice to design and implement a capability based access control system for the EU FP7 IoT@Work project are to address the following needs:

- manage a significant number of *subjects* (suppliers, maintainers, etc.) belonging to different companies that need to access resources in the production plant;
- assure compliance with the *Principle of Least Authority*;
- support easy-to-use rights delegation (and delegation control) assuring, at the same time, full auditability of resource access;
- offload management to face external subjects dynamics.

The next sections quickly describe our CapBAC systems and its features.

4. THE CapBAC SYSTEM

The CapBAC described in the following borrows ideas and approaches from previous works extending and adapting them to address IoT requirements and specifically the EU FP7 IoT@Work project's ones. Indeed in IoT@Work, the CapBAC system is used as the access control system of the *ENS* (*Event Notification Service*) automation middleware that acts as a common collector of events acquired from disparate sources in an industrial plant and dispatched, in a controlled way, to a set of listeners (events' consumers). Our CapBAC system provides the following features:

- **delegation support**: a *subject* can grant access rights to another *subject*, as well as grant the right to further delegate all or part of the granted rights. The delegation depth can be controlled at each stage;
- **capability revocation**: capabilities can be revoked by properly authorized *subjects*, therefore solving one of the issues of capability based approaches in distributed environments. Thanks to the delegation feature, a given access capability can have capability sub-trees where each child capability grants some or all parent capability delegable rights to another *subject*. The owner of an access capability has the right to revoke one or more of his/her/its descendant capabilities;
- **information granularity**: a capability can even specify dynamic adaptation of the granted rights (e.g. specify a "level of details" for a read access right on a specific piece of information). In this way the service provider can refine its behavior and the data it has to provide;
- **XML representation**: access capabilities (Capability Tokens) are represented by digitally signed XML files;
- **SAML/XACML based**: we use/extend SAML/XACML for capability token's elements.

As an example, Figure 2 depicts the usage of access capabilities to control access to Bob's car information and services (e.g.: car's location in Figure 2 (a), car's engine status in Figure 2 (b)). The involved subjects are: Bob Smith, the car's owner; Alice Cooper, Bob's wife (interested in having information on Bob's car location); the City Traffic Management Service (interested in monitoring cars' location); the Car Manufacturer's Maintenance Service (the application service in charge of monitoring engines status); Dave Jones (manager of the car's manufacturer Maintenance Service). As depicted in the figure, Bob provides access capabilities to Alice, the City Traffic Management Service and to Dave. In particular: Alice's access capability grants her the right to query Bob's car location with high precision (see Access Capability $\alpha 2$); the City Traffic Management Service's access capability grants the same right but with a *Block Level* precision (see Access Capability $\alpha 1$); Dave's access capability grants him Query and Change rights on Bob's car engine status (see Access Capability $\beta 1$ in the figure). Additionally Dave's access capability grants him the right to further delegate his rights. As indicated in Figure 2 (b), Dave, on the basis of his access capability, has created an additional capability (Access Capability $\beta 2$) for the car manufacturer's maintenance service that periodically monitors Bob's car engine status providing it a subset of his rights.

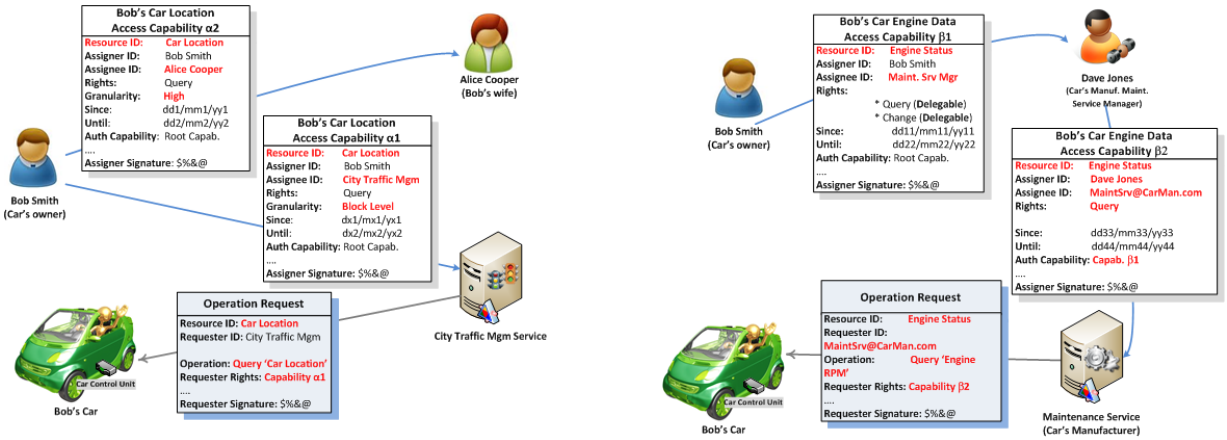

Figure 2. CapBAC – Examples of potential scenarios.

Figure 2 also shows two service requests submitted to Bob's car control unit. Each request states what is requested to the control unit and includes: the access capability granting the access rights on the resource the request asks to act upon, the requestor's identity and the proof of identity ownership. Bob's car control unit has therefore all the elements to evaluate if the access request is acceptable or not. As a final remark it is worthwhile to highlight that the service provider (e.g. Bob's car control unit in the example) has full visibility of the authorization chain and, therefore, each subject is fully accountable.

4.1 CapBAC Functional Elements

To fully address even complex environment's needs, like the ones targeted by our IoT@Work project, our CapBAC architecture envisages a set of functional elements (sketched in Figure 3), which have not to be necessarily deployed in all environments.

The uniquely identifiable and actable upon resource (e.g. a RESTful resource) on which rights are granted (*Acme Ltd Service A* in the figure) is the **access capability object**. It can be a specific piece of information (e.g. your heartbeats as measured by a sensor and made available via your smart phone), an application service (e.g. Alice's mailbox IMAP service) or a mix of services.

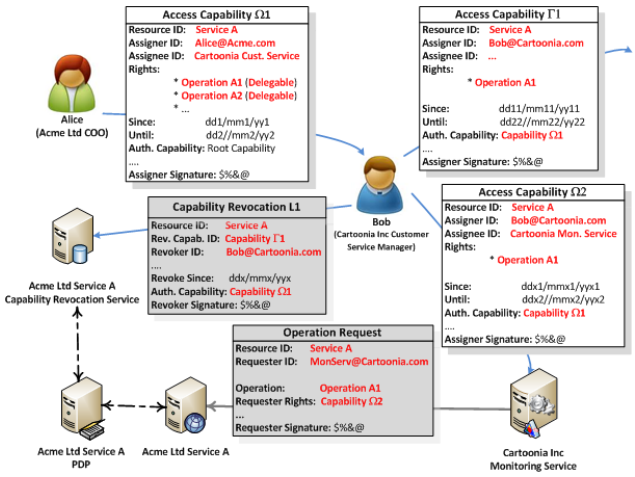

Figure 3. CapBAC- functional elements.

The **authorization capability** details the granted rights (including the delegation rights and their delegation depths), the resource on which those rights can be exercised, the issuer's and grantee's identities (currently they are simple identifiers tied to X.509 certificates; the only constraint is that these identifiers can be univocally tied to the access capability proof of ownership when an access request is actually submitted), as well as additional information (e.g. capability validity period – an authorization capability is valid as specified within the capability itself or until it is explicitly revoked, XACML conditions, access capability digital signature, etc.). As stated, a capability is a communicable object and therefore it can be provided to the subject (grantee) using any communication mean (see *Access Capability Ω1*, *Access Capability Ω2*, and *Access Capability Γ1* in Figure 3).

The **capability revocation** revokes one or more capabilities (i.e. a single specific capability, or a specific capability together with its descendants, or all descendants of a specific capability). Like a capability, it is a communicable object a subject having appropriate rights creates to inform the service provider that specific capabilities have to be considered no more valid (see *Capability Revocation L1* in Figure 3).

In the CapBAC system, a **service/operation request** (See *Operation Request* in Figure 3) is the service request expressed and communicated as envisaged by the provided service with the only additional characteristics to refer or include, in an unforgeable way, a capability. For example, for a RESTful service, an HTTP GET request has to include the capability and its proof of ownership to use our access control mechanism.

Such service/operation requests are managed by the **resource manager** (see *ACME Ltd Service A* in Figure 3) that is the service provider in charge of managing the identified resource and of providing the requested service. From a CapBAC point of view, it is in charge of validating the capability in the service request and the congruence of the request against the provided access capability. The resource manager, as normally happens, has to act like an *XACML Policy Enforcement Point* (*PEP*) taking into account the validation outcomes of the *Policy Decision Point*.

The **Policy Decision Point** (**PDP** – see *ACME Ltd Service A PDP* in Figure 3) is a resource-agnostic service in charge of managing resource access request validation and decision. More specifically, in our CapBAC system it is in charge of validating the access rights granted in the capability against local policies and checking

the revocation status of the capabilities in the delegation chain. As for an XACML PDP the outcome of this evaluation is an *Allow* or *Deny*.

Finally, the **Capability Revocation Service** is in charge of managing capability revocations (see *ACME Ltd Service A Capability Revocation Service* in Figure 3). Its work, therefore, envisages both the validation of the received capability revocations, as well as updating PDP capabilities database and access policies accordingly.

As evident from the above description, the CapBAC system main differences with traditional access management systems relies on these additional elements: access capability, capability revocation and revocation service. These elements add flexibility to the authorization framework providing more granularity, scalability, easy support for access rights delegation, and full accountability of the authorization chain. The main drawback of the CapBAC system is that it requires issuing capabilities to all involved subjects, even if this management issue can be easily split among many subjects and be deferred until a new subject actually need to have access to the managed *objects*.

4.2 CapBAC and Consumer's Scenarios

A capability based access control system, as highlighted, is based on the provision to *subjects* that need access to a resource of a specific digital token. This token directly identifies the granted *subject* and the granted rights. Additionally the granted rights can be tailored for a managed resource. The only constraint is that the granted rights are known to the service provider. These peculiarities well addresses the dynamicity and heterogeneity of consumer's scenarios (e.g. domotics, e-Health) where a user, starting from a very basic set of information about a service (information that could be embedded in smart devices or appliances by the manufacturers or installers) can easily catch the CapBAC basic concept "*if someone has to access your appliance or device you have to provide him/her an access token. It is up to you to actually define what rights you grant via the token to the subject and for how long*" and act accordingly. For a householder, for example, is immediate and natural to manage temporary delegation of rights for his/her home appliances providing a CapBAC access token to his/her neighbor when he/she has to go on holidays.

The processing power and storage or communication resources required by our CapBAC system are compatible with commercial smart phones or tablets or any device that is able to process XML files of some Kbytes and manage X.509 certificate based digital signature. For less powerful devices the processing and storage requirements could be further reduced using an EXI (Efficient XML Interchange) based encoding and ECC based cryptography.

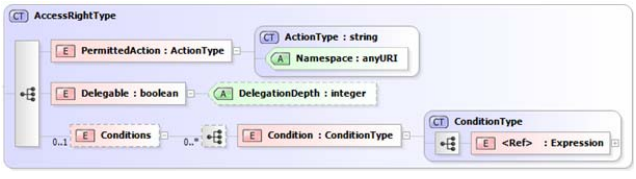

Figure 4. CapBAC XML schema.

Our CapBAC schema, as shown in Figure 4, envisages the possibility to embed in an access capability also *Conditions* defined according to the XACML *ConditionType* indications, therefore enabling the possibility to add flexibility, on a token by token basis, to the access control mechanism. Indeed, these *Conditions* can be used for example to specify the granularity or precision of the provided information (as discussed for Figure 2) or to add constraints like the usability of an access capability only when the user is in a specific environment or condition.

5. CONCLUSIONS AND FUTURE WORK

The CapBAC system offers the advantages listed in Chapter 3.2 and it is being implemented in Java as a set of libraries, tools and services. Some modules are OSGi based to be deployable on small devices. We have a first version of the required libraries and tools, as well as of the CapBAC PDP and the Revocation Service. These two services are Java web applications, providing an AJAX UI for their management and REST APIs. As stated the system is used to manage access control to the IoT@Work ENS middleware. Some CapBAC features are also being evaluated in the EU FP IoT6 project (http://www.iot6.eu), in particular by the University of Murcia in a CoAP based context.

In the immediate future we plan to revise the system taking into account the feedbacks coming from our IoT@Work and IoT6 validations.

The CapBAC system is being made available openly (Apache License V2) in order to both speed up its adoption and to refine and enhance it.

6. ACKNOWLEDGMENTS

This work was financially partially supported by the European FP7 EU Project IoT@Work under grant number ICT-257367.

7. REFERENCES

[1] Issarny, V., Georgantas, N., Hachem, S., Zarras, A., Vassiliadist, P., Autili, M., Gerosa, M., and Hamida, A. 2011. Service-oriented middleware for the Future Internet: state of the art and research directions. *Journal of Internet Services and Applications* 79, 1 (2011), 23-45. DOI= http://dx.doi.org/10.1007/s13174-011-0021-3

[2] Christophe, B., Boussard, M., Lu, M., Pastor, A., and Toubiana, V. 2011. The web of things vision: Things as a service and interaction patterns. *Bell Labs Technical Journal*, 16, 1 (June 2011), 55-61. DOI= http://dx.doi.org/10.1002/bltj.20485

[3] Mehra, P. 2012. Context-Aware Computing: Beyond Search and Location-Based Services. *IEEE Internet Computing*, 16, 2 (March-April, 2012), 12-16

[4] Trappeniers, L., Roelands, M., Godon, M., Criel, J., and Dobbelaere, P. 2009. Towards Abundant DiY Service Creativity Successfully Leveraging the Internet-of-Things in the City and at Home. In *Proceedings of the 13th Int. Conf. on Intelligence in Next Generation Networks* (Bordeaux, France, October 26 - 29, 2009). ICIN 2009.

[5] Uckelman, D., Harrison, M., and Michahelles, F. (eds.) 2011. *Architecting the Internet of Things*. Springer-Verlag Berlin Heidelberg

[6] Butler W. Lampson. 1974. Protection. *SIGOPS Oper. Syst. Rev.* 8, 1 (January 1974), 18-24. DOI= http://doi.acm.org/10.1145/775265.775268

[7] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R. 2001. Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.* 4, 3 (August 2001), 224-274. DOI= http://doi.acm.org/10.1145/501978.501980

[8] Sandhu, R. Ferraiolo, D., and Kuhn, R. 2000. The NIST model for role-based access control: towards a unified standard. In *Proc. of the 5th ACM workshop on Role-based access control* (Berlin, Germany, July 26 - 28, 2000). RBAC '00. ACM, New York, NY, USA, 47-63. DOI= http://doi.acm.org/10.1145/344287.344301

[9] Simon S. Y. Shim, Geetanjali Bhalla, and Vishnu Pendyala. 2005. Federated Identity Management. *Computer* 38, 12 (December 2005), 120-122. DOI= http://dx.doi.org/10.1109/MC.2005.408

[10] eXtensible Access Control Markup Language Version 3.0, *OASIS XACML* v. 3.0, August 2010

[11] Yee, K. P. 2003. Secure interaction design and the principle of least authority. In *Proc. of the 21st Int. Conf. on Human Factors in Computing Systems – Workshop on Human-Computer Interaction and Security Systems*, (Ft. Lauderdale, FL, USA, April 6, 2003). CHI 2003. ACM, New York, NY, USA

[12] Saltzer, J.H. and Schroeder, M.D. 1975. The protection of information in computer systems. In *Proceedings of the IEEE*, 63, 9, (Sept. 1975), 1278- 1308. DOI= http://dx.doi.org/10.1109/PROC.1975.9939

[13] Skinner, G. D. 2009. Cyber Security Management of Access Controls in Digital Ecosystems and Distributed Environments. In *Proc. 6th Int. Conf. on Information Technology and Applications* (Hanoi, Vietnam, November 9 - 12, 2009), ICITA 2009.

[14] Fang, L., Gannon, D., and Siebenlist, F.. 2005. XPOLA – An Extensible Capability-based Authorization Infrastructure for Grids. In *4th Annual PKI R&D Workshop*, April 2005

[15] Adams, A., and Sasse, M. A. 1999. Users are not the enemy. *Commun. ACM* 42, 12 (December 1999), 40-46. DOI= http://doi.acm.org/10.1145/322796.322806

[16] Beznosov, K., Inglesant, P., Lobo, J., Reeder, R., and Zurko, M. E. 2009. Usability meets access control: challenges and research opportunities. In *Proc. of the 14th ACM Symposium on Access control models and technologies* (SACMAT '09). ACM, New York, NY, USA, 73-74. DOI= http://doi.acm.org/10.1145/1542207.1542220

[17] Mazurek, M. L., Arsenault, J. P., Bresee, J., Gupta, N., Ion, I., Johns, C., Lee, D., Liang, Y., Olsen, J., Salmon, B., Shay, R., Vaniea, K., Bauer, L., Faith Cranor, L., and Ganger, G. R. 2010. Access Control for Home Data Sharing: Attitudes, Needs and Practices. In *Proc. of the 28th Int. Conf. on Human Factors in Computing Systems* (Atlanta, GA, USA, April 10 - 15, 2010). CHI 2010. ACM, New York, NY, USA, 645-654. DOI= http://doi.acm.org/10.1145/1753326.1753421

[18] Kostiainen, K., Rantapuska, O., Moloney, S., Roto, V., Holmstrom, U., and Karvonen, K. 2007.Usable Access Control inside Home Networks. *IEEE Int. Symposium. on World of Wireless, Mobile and Multimedia Networks* (Helsinki, Finland, June 18-21, 2007). WoWMoM 2007. IEEE Computer Society, Washington, DC, USA, 1-6. DOI= http://doi.acm.org/10.1109/WOWMOM.2007.4351809

[19] Dennis, J. B., and Van Horn, E. C. 1965. *Programming Semantics For Multiprogrammed Computations*. MIT, Tech. Report MIT/LCS/TR-23, 1965.

[20] Levy, H. 1984. *Capability-Based Computer Systems*. Digital Press, Bedford, Massachusetts, 1984.

[21] Tanenbaum, A. S., Mullender, S. J., and van Renesse, R. 1986. Using Sparse Capabilities in a Distributed Operating System. In *Proc. 6th Int. Conf. on Distributed Computing Systems* (Cambridge, MA, USA, May 19 - 13, 1986) ICDCS 1986, 558–563.

[22] *SPKI Certificate Theory*, IETF RFC 2693, September 1999

[23] ITU-T Recommendation X.509: *Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks* (also known as ISO/IEC 9594-8), ITU-T Recommendation X.509, November 2008.

[24] *An Internet Attribute Certificate Profile for Authorization*, IETF RFC 3281, April 2002.

[25] Balfanz, D., Durfee, G. E., and Smetters, D. K. 2005. Making the impossible easy: usable PKI. In *Security and Usability*, Chap. 16 edited by L. Cranor and S. Garfinkel, O'Reilly, 2005.

[26] Lackorzynski, A. and Warg, A. 2009. Taming subsystems: capabilities as universal resource access control in L4. In *Proc. of the 2nd Workshop on Isolation and Integration in Embedded Systems* (Nürnberg, Germany, March 31, 2009). IIES '09. ACM, New York, NY, USA, 25-30. DOI= http://doi.acm.org/10.1145/1519130.1519135

[27] Li, J., and Karp, A. H. 2007. Access control for the services oriented architecture. In *Proc. of the 2007 ACM workshop on Secure Web Services* (Fairfax, VA, USA, November 2, 2007). SWS '07. ACM, New York, NY, USA, 9-17. DOI= http://doi.acm.org/10.1145/1314418.1314421

[28] Karp, A. H. 2006. Authorization-Based Access Control for the Services Oriented Architecture. In *Pro. of the 4th Int. Conf. on Creating, Connecting and Collaborating through Computing* (Cambridge, MA, USA, January 28 - 29, 2005). C5 '06. IEEE Computer Society, Washington, DC, USA, 160-167. DOI= http://dx.doi.org/10.1109/C5.2006.9

[29] Karp, A. H., Haury, H., and Davis, M. H. 2009. *From ABAC to ZBAC: The Evolution of Access Control Models*. HP Laboratories, Tech. Report HPL-2009-30, February 2009

[30] Karp, A. H., and Li, J. 2010. Solving the Transitive Access Problem for the Services Oriented Architecture. In *Proc. 2010 Int. Conf. on Availability, Reliability, and Security* (Krakow, Poland, February 15 - 18, 2010).ARES '10. IEEE Computer Society, Washington, DC, USA, 46-53. DOI= http://doi.ieeecomputersociety.org/10.1109/ARES.2010.34

[31] Miller, M., Ka-Ping Yee, and Shapiro, J. 2003. *Capability Myths Demolished*. Systems Research Laboratory, Johns Hopkins University, Tech. Report SRL2003-02, 2003

[32] Factor, M., Naor, D., Rom, E., Satran, J., and Tal, S. 2007. Capability based Secure Access Control to Networked Storage Devices. In *Proc. of the 24th IEEE Conf. on Mass Storage Systems and Technologies* (San Diego, CA, USA, September 24-27, 2007).MSST '07. IEEE Computer Society, Washington, DC, USA, 114-128. DOI= http://dx.doi.org/10.1109/MSST.2007.6

[33] Hardy, N. 1988. The Confused Deputy: (or why capabilities might have been invented). *SIGOPS Oper. Syst. Rev.* 22, 4 (October 1988), 36-38.