XACML for Building Access Control Policies in Internet of Things

Hany F. Atlam[1, 2], Madini O. Alassafi[1], Ahmed Alenezi[1], Robert J. Walters[1] and Gary B. Wills[1]
[1]*Electronic and Computer Science Dept., University of Southampton, University Road, SO17 1BJ, Southampton, U.K.*
[2]*Computer Science and Engineering Dept., Faculty of Electronic Engineering, Menoufia University, 32952, Menouf, Egypt*

Keywords: Internet of Things, Access Control, XACML, Policy Language, Adrbac, Access Policies.

Abstract: Although the Internet of things (IoT) brought unlimited benefits, it also brought many security issues. The access control is one of the main elements to address these issues. It provides the access to system resources only to authorized users and ensures that they behave in an authorized manner during their access sessions. One of the significant components of any access control model is access policies. They are used to build the criteria to permit or deny any access request. Building an efficient access control model for the IoT require selecting an appropriate access policy language to implement access policies. Therefore, this paper presents an overview of most common access policy languages. It starts with discussing different access control models and features of the access policy. After reviewing different access policy languages, we proposed XACML as the most efficient and appropriate policy language for the IoT as it compatible with different platforms, provides a distributed and flexible approach to work with different access control scenarios of the IoT system. In addition, we proposed an XACML model for an Adaptive Risk-Based Access Control (AdRBAC) for the IoT and showed how the access decision will be made using XACML.

1 INTRODUCTION

The Internet of Things (IoT) is currently a hot area that attracts the attention of both academia and commercial organizations. It has the ability to dramatically change our lives and businesses for the better especially with its integration with Cloud computing (Atlam et al., 2017a).

The IoT brought unlimited benefits, however, it also brought many security issues. One of the main elements to address IoT security issues is the access control. The main purpose of the access control is to reject unauthorized users and limit operations of authorized users using a certain device (Atlam et al., 2017c). One of the essential elements of any access control model is access control policies. These policies are implemented using an access policy language.

This paper provides an overview of common access policy languages. It starts with showing different access control models, then providing a review of access policy languages to select the most appropriate one to implement access control policies in the IoT. After reviewing existing policy languages, eXtensible Access Control Markup Language (XACML) is proposed to the best choice to build access control policies for the IoT as it is flexible and powerful policy language. In addition, we proposed an XACML model for an Adaptive Risk-Based Access Control (AdRBAC) for the IoT and have shown how the access decision will be made using XACML.

The rest of this paper is organized as follows: Section 2 presents access control; Section 3 provides an overview of different access control models; Section 4 introduces access control policies; Section 5 presents access control policy language for IoT; Section 6 discusses XACML model and its structure, Section 7 presented proposed XACML model, and finally Section 8 is the conclusion.

2 ACCESS CONTROL

Access control is one of the security technologies to protect system resources by preventing unauthorized access to resources and restricting legitimate user's access according to their privileges (Hu et al., 2006).

The access control model is implemented at different levels in many areas such as operating system and database management system. Any access control is consisting of five factors as follows:

Atlam, H., Alassafi, M., Alenezi, A., Walters, R. and Wills, G.
XACML for Building Access Control Policies in Internet of Things.
DOI: 10.5220/0006725102530260
In *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security (IoTBDS 2018)*, pages 253-260
ISBN: 978-989-758-296-7

1. ***Subjects:*** Active entities in the form of users and processes that request the access to objects.
2. ***Objects***: passive entities containing information being accessed by subjects.
3. ***Actions***: An operation to be performed on a certain object (read, write, execute, etc.).
4. ***Privileges***: Authorizations permissions to perform certain actions on certain objects.
5. ***Access policies***: The set of rules that determine the access decision whether accepted or denied.

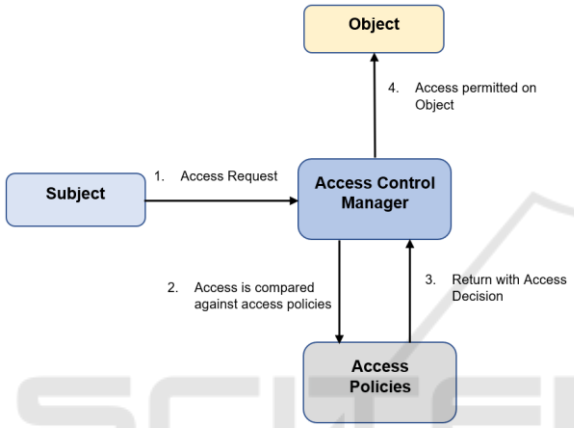

Figure 1: Main elements of an access control model.

Figure 1 shows the flow of an access control operation. It starts when a subject/user send an access request to the access control manager to access a certain object. The access control manager uses the user details such as username, password, identity card, and Biomatrices then compares it against the access control policies to determine the access decision. The decision will be either accepted or rejected (Suhendra, 2011). If the access is accepted, the access control manager will allow the user to access the object, while if the access is rejected, the access control manager will terminate the session after sending warning message regarding insufficient credentials (Liu et al., 2012).

3 ACCESS CONTROL MODELS

The main purpose of the access control is to determine if a user is authorized to access a resource, data, service and determine the access decision whether accepted or denied. Access control model is an essential part of any information management system to protects system resources from unauthorized access and ensures achieving confidentiality, integrity, and availability to system users (Atlam et al., 2017b). This section provides an overview of common access control models.

3.1 Access Control List

Access control was traditionally implemented with a matrix table called Access Control Matrix (ACM), where each row and column is composed of a subject and object respectively. Each record represents a set of access rights for the corresponding subject (N.Mahalle et al., 2013).

After that, the Access Control List (ACL) appeared. ACL is a list of a certain object which contains legitimate subjects along with their access rights (Hu et al., 2006).

3.2 Discretionary Access Control

Discretionary Access Control (DAC) model was designed for multi-user databases and systems with a few previously known users.

DAC grants access depending on the user identity and authorization, which is defined for open policies. The owner of the resource can grant the access to any user. It is called discretionary as it provides the flexibility to allow the users to pass their access permissions freely to other users. Therefore, all the system resources are under full control from the user (Atlam et al., 2017b; Janak et al., 2012).

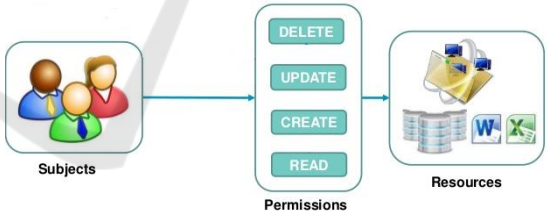

Figure 2: DAC access control model (Janak et al., 2012).

3.3 Mandatory Access Control

In the Mandatory Access Control (MAC), each object is assigned a label which specifies security privileges of the object based on the sensitivity of the information in the object, and each subject is assigned a label that specifies which object the requester can access (Bugiel et al., 2013; Hulsebosch et al., 2005). DAC model provides security measures where a user can only perform tasks related to its privileges.

The MAC model is concerned with confidentiality and integrity of information, so it mainly used in military and government applications. In MAC, the

security policy is controlled by a security policy administrator and the user does not have the capability to override it (Zhu and Jin, 2007).

3.4 Role-based Access Control

Role-Based Access Control (RBAC) is a widely accepted model in almost all large enterprises (Bijon et al., 2013). RBAC model is consists of three elements: users (subjects requesting access), roles (collection of permission) and operations (actions on target resource). Access permissions are related to roles and the appropriate role is granted to the user. A single user can be associated with one or more roles, and a single role can include one or more user. RBAC provides a classification of users based on their roles (Atlam et al., 2017c; Kumar et al., 2002).

The RBAC model restricts access to objects based on the subject's role rather than their identifications. Roles are allocated to subjects according to their clearance, qualification, and responsibilities inside the organization. A set of permissions are grouped together to form a role. A user can be allocated to different roles and the role can be assigned to different users. The RBAC model might have many users, each user will be assigned to a specific role or may be assigned to multiple roles and each role consists of a set of permissions/rights. An example of RBAC in a hospital, where doctors can both read and write prescriptions, whereas pharmacists are limited to read prescriptions only.

3.5 Attribute-based Access Control

Attribute-Based Access control (ABAC) is a dynamic access control model incorporating key factors that called attributes that belong to the entities of subjects, objects, actions, and environment conditions (Jin et al., 2012).

ABAC is an access control model that grants or denies access to objects based on the assigned attributes of the subject, object, environmental conditions and the set of access control policies specified based on those attributes and conditions. Therefore, access policies are created without the need for direct reference to subjects or objects (Liu et al., 2016).

The use of attributes has a remarkable capacity for dealing with an unlimited number of subjects without any modifications to existing policies. Therefore, achieving less maintenance and overhead (Atlam et al., 2017d).

3.6 Cryptographic RBAC

Cryptographic RBAC model is presented to provide secure data outsourcing and an effective resource management in RRBAC systems. With the presence of the cloud computing which provides unlimited benefits to users and applications. Cloud data centres are distributed geographically and anonymously, therefore, once the user uploads their data to the cloud, users lose control of their data (Zhu et al., 2010a).

In addition, the cloud service providers can access user's data even if it's not allowed to do so. Therefore, access control policies should be implemented to allow only actual data owners to access their data (Zhou et al., 2014a). The Cryptographic RBAC comes into play.

Cryptographic RBAC, as stated by its name, integrate two concepts, which are the RBAC and Cryptographic mechanism. This access control model allows the users to encrypt the data before storing it in the cloud so that only data owners will have the decryption key to decrypt the data. The encrypted data is specific to roles, therefore only users who are part of these roles can decrypt the data with their own secret key (Zhou et al., 2014b; Zhu et al., 2010b).

4 ACCESS CONTROL POLICIES

The access control is used to manage every access to the system resources to ensure that all and only authorized users are allowed to access. To do so, access control is based on access rules which define accesses to be allowed or to denied. An administrative policy is therefore needed to regulate the specification of such rules to specify who can add, delete, or modify system resources. Administrative policies are one of the most important aspects of access control.

According to IBM, the access control policies are defined as "*a set of conditions that, after they have been evaluated, determine access decisions. The conditions are a combination of attributes, obligations, authentication policies, and a risk profile*" (Molloy et al., 2011).

According to VimercatiS et al. (2007), An access control policy should have these features:

- ***Efficiency****:* The efficiency of the access policy is always a serious issue. Therefore, providing an efficient and simple approach to allow or deny the access is a key point in any access control model.
- ***Simplicity***: One of the main issues in defining a policy language is the ability to enable

expressiveness and flexibility with guaranteeing easiness of use and applicability. An access control language should be based on a high-level formulation of the access control rules, possibly close to natural language formulation.

- ***Expressibility***: The language used to implement the policy should be expressive so that it can suit all the data owner's needs. To do so, many of new language designs depend on perceptions and methods from logic, specifically from logic programming.
- ***Policy combination and conflict-resolution:*** with the existence of multiple modules of different authorities or different domains for the specification of access control rules, the access control system should provide a means for users to specify how the different modules should interact.

4.1 Privacy Policy Languages

Access policy languages provide many advantages in several stages of implementing access control policies. It is called privacy policy languages; these languages are designed to express the privacy controls that both organizations and users want to express. Most of the privacy policy languages were designed for specific purposes with specific features and characteristics (Kumaraguru et al., 2007).

There are serval access control languages. This section provides an overview of most common languages.

4.1.1 XACML

XACML is an access control policy language that provides two-way communication for the request and response of an access request (Oasis, 2005). It is the most popular policy language that can work with different access control models. It comprises of a group of standard XML components and specifies standard extension points for individual rules, data types and procedures. XACML is created by OASIS standard, there are many implementations and versions for XACML (Rissanen, 2010).

4.1.2 XACL

The XML Access control language (XACL) was established by the IBM Tokyo Research Laboratory in 2000 (Bauer et al., 2004). It based on XML and used to provide access control policies that could be enforced on the access to XML documents. XACL is used primarily for specifying object-subject-condition policies in which a subject can have an identity, or a rule and Objects can address single elements in an XML document. Actions can include read, write, create and delete. The right to carry out an action can be bound to provisions like auditing, verification of a digital signature, encryption, XSL transformations, or simple additional actions (Hada and Michiharu Kudo, 2000).

4.1.3 APPEL

The Adaptable and Programmable Policy Environment and Language (APPEL) has been established in the ACCENT project at the University of Stirling in 2013 (Turner et al., 2014). It originally used as means of providing policies for automatic telephone call control. Due to its extensibility and domain-independency, it can be used in different domains. APPEL provides a simple but expressive syntax that is intended to be usable by unprofessional users while providing means for experts to describe of complex details of a system. A policy rule contains triggers, conditions and actions. The specific names used in the rule contents are defined according to the domain of each use case (Bugiel et al., 2013).

4.1.4 P3P

The Platform for Privacy Preferences (P3P) has been developed as a W3C standard for the expression of web user's privacy preferences and data collection policies of a service provider (W3C, 2006). A preference or policy is used to specify the purpose of collecting certain data items, who will receive the data when will the collected data be reserved. Agents could be used by users to automatically extract the information of the data collection policies of a service provider and match it against her preference. Based on the outcome of a preferences/policy match, they may proceed in using a service, and hence, share the information mentioned in the policy. Or the user decides not to use a service. P3P is used to help users to understand and be mindful of the process of collecting privacy-relevant data (Bugiel et al., 2013).

4.1.5 EPAL

The Enterprise Privacy Authorization Language (EPAL) has been developed by IBM to specify enterprise privacy policies on collected data in an enterprise (Ashley et al., 2003). It provides means of administrating data handling practices in an enterprises' IT systems. It allows forming positive and negative authorization rights. EPAL policies specify hierarchical data classes of the collected data, data user classes, data usage purposes, sets of (privacy) actions on the collected data, obligations, and conditions (Ashley et al., 2003).

5 ACCESS POLICY LANGUAGE FOR IoT

Like all new technologies, The IoT has many security issues. Authentication and access control models are the essential elements to address these security issues. The main purpose of the access control is to reject unauthorized users and limit operations of authorized users using a certain device (Atlam et al., 2017c, 2013).

One of the major points in creating an efficient access control model for the IoT is to build a flexible and distributed access control policies. This can be done by choosing the most appropriate policy language to implement access control policies.

After providing the summary of most common policy languages in the previous section, we believe that XACML is the most appropriate policy language to implement access control policies in the IoT environment. This is because XACML has many advantages over other access control policy languages. Advantages of XACML are as follows:

- ***Standardized***: XACML is a standard policy language, that has been revised and reviewed by many experts and users. In addition, XACML is the most popular policy language and widely deployed in many access control models, therefore, it will be easier to interoperate with other applications using the same standard language.
- ***Generic***: XACML policy can be deployed on different platforms. One policy can be written to be used by many different applications, and when one common language is used, policy management becomes much easier.
- ***Distributed***: XACML policy can be written to refers to other policies kept in remote locations. therefore, different users or groups can manage policies and XACML can properly integrate the results from these different policies into one decision.
- ***Powerful***: XACML supports a wide diversity of data types, functions, and rules about combining the results of different policies. In addition, there are already standards groups working on extensions and profiles that will hook XACML into other standards like SAML and LDAP, which will increase the number of ways that XACML can be used.
- ***Flexibility:*** XACML is based on ABAC model which is a dynamic model which will be good for the IoT which is characterized as a dynamic system. XACML can adapt to different changes, situations, and conditions in the IoT environment.

Since we selected XACML to be the policy language to build access control policies in IoT systems, the next section will provide more details about XACML components and how it works.

6 XACML

XACML is a popular standard for specifying access control policies which determine the access decision whether to permit or deny the access. XACML is considered one of the most promising policy language dealing with dynamic and complex systems. It is broadly accepted by a vast of majority of experts, communities and organizations since it is compatible with most access control models such as ACL, RBAC, and ABAC (Rissanen, 2010).

6.1 XACML Architecture

The basic components of XACML architecture are as follows:

- ***Policy Administration Point (PAP):*** is an interface for searching policies. It retrieves the policies applicable to a given access request and returns them to the PDP module (Oasis, 2005).
- ***Policy Enforcement Point (PEP)****:* receives the access request in a simple format and moves it to the Context Handler. Likewise, when a decision has been made by the decision point, PEP applies the access decision that it receives from the Context Handler (Liang Chen, Luca Gasparini, 2013).
- ***Policy Decision Point (PDP):*** takes an access request and interacts with the PAP that encapsulates the information needed to identify the applicable policies. It then evaluates the request against the applicable policies and returns the authorization decision to the Context Handler (Oasis, 2005).
- ***Policy Information Point (PIP):*** provides attribute values about the subject, resource, and action. It interacts directly with the Context Handler (OASIS, 2003).
- ***Context Handler***: translates access requests into a simple format. Basically, it acts as a bridge between PDP and PEP and it is responsible for retrieving attribute values needed for policy evaluation (Gasparini, 2013).
- ***Environment***: provides a set of attributes that are relevant to make an authorization decision

and are independent of a particular subject, resource, and actions (Oasis, 2005).

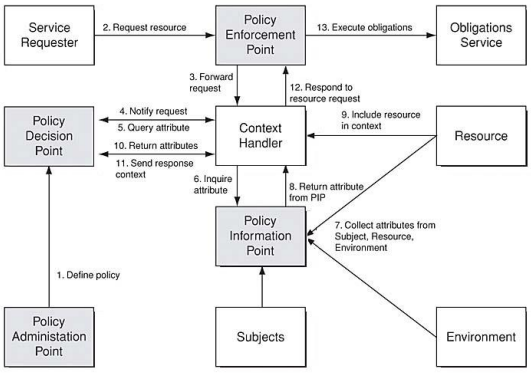

Figure 3: XACML architecture.

As shown in Figure 3, the data follow diagram of all 13 steps is as follows:

1. Policies used by PDP are established by PAP.
2. The access requester sends an access request to PEP, which in turn passes it in its native form to context handler together with attributes when necessary, as shown in step 3.
4. Context handler makes an XACML request to PDP with required attributes.
5. PDP requests additional attributes to be retrieved in the 5-10 steps from context handler.
11. PDP evaluates the policies and then returns the response context including the authorization decision to context handler.
12. Context handler sends to the context converted into the native response format for PEP.
13. PEP conduct obligations and if the authorization is granted, the access requester will be permitted to access the resource, otherwise, PEP will deny the access.

6.2 XACML Policy Structure

A policy has prevention and detection capabilities which are used to make the access decision. The main components of the policy are Rule, Policy, and Policy set. Rules are the minimum components of a policy which is consists of three elements: target, effect, and condition (Westphall and Schmitt, 2016).

The target is a set of decision requests, which plays an important role in narrowing down multiple policies in place into only applicable ones through target match with its inner components. The effect is the intended result (permit or deny) to be provided when a rule is satisfied. The condition is the function to be performed when a rule target is applicable, leading to a result of 'True' or 'False'. The policy is a set of rules grouped together using a rule-combing algorithm with its selective combining parameters (Liang Chen, Luca Gasparini, 2013; Oasis, 2005).

6.3 XACML Policy Evaluation

The XACML engine evaluates a given XACML request against multiple policies independently. The XACML engine is a software component in XACML which require two inputs and return one output. The two inputs are XACML policies and access request, while the output is the access decision (Jayasekara, 2011), as shown in Figure 4.

According to (Jayasekara, 2011), the output of a given access request is one of the following:

- ***Permit:*** Request is authorized to carry out operation/actions on system resources.
- ***Deny:*** Request is not authorized to perform operations requested.
- ***Not Applicable***: XACML engine could not find any applicable policies regarding the access request and the requester.

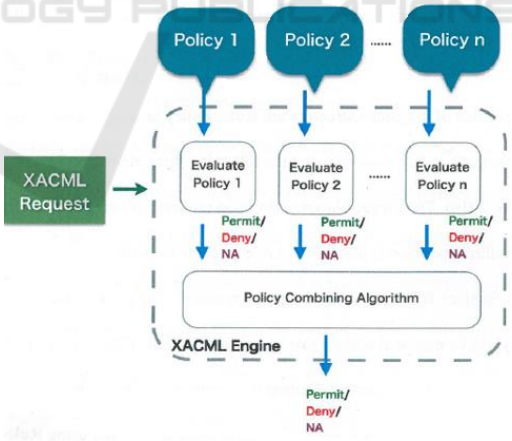

Figure 4: Policy evaluation (Jayasekara, 2011).

6.4 XACML Target Evaluation

To provide more efficiency and ease of evaluation, the target element in each policy plays an important role in narrowing down multiple policies into only applicable ones to a given XACML request (Jayasekara, 2011).

The XACML engine does not necessarily evaluate all stored policies. It is done by comparing four sub-elements attributes (subject, resource, action, and environment) in the target element of a policy with the corresponding attributes in an XACML request, as shown in Figure 5.

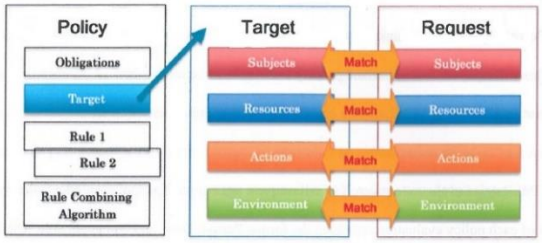

Figure 5: Target evaluation.

7 PROPOSED XACML MODEL

We proposed an Adaptive Risk-Based Access Control (AdRBAC) model for the IoT (Atlam et al., 2018, 2017c). The proposed AdRBAC model has four inputs; user/agent context, resource sensitivity, action severity and risk history. These inputs/risk factors are used to estimate the security risk associated with each access request. The estimated risk value is then compared against the risk policies to make the access decision (Atlam et al., 2018, 2017c).

Specifying the appropriate risk policy that will be used with the proposed risk model was done using XACML. We proposed XACML model to build Access policies of the Proposed AdRBAC, as shown in Figure 6.

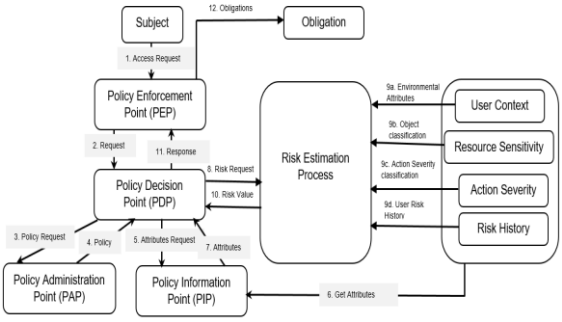

Figure 6: Proposed XACML model for the proposed Risk-based access control model.

The flow of the access decision process can be tracked in Figure 6, that starts with the user (subject) who sent an access request. This access request is processed through the system and the risk value regarding values of user context, resource sensitivity, action severity, and risk history is estimated and the access decision is made whether granting or denying the access.

8 CONCLUSION

The proliferation of interconnected IoT devices and applications brought various security and privacy challenges. The access control is one of the main element to address these issues. One of the key elements of access control model is access control policies. Building an efficient access control model for the IoT require choosing the most appropriate access policy language that can provide scalability and flexibility for the IoT system. This paper has presented a review of common privacy policy languages to determine the most efficient language to be used for the IoT. XACML, XACL, APPEL, P3P, and EPAL policy languages have presented with discussing main structure and components of XACML. We conclude that XACML is the best choice for the IoT as it compatible with different platforms, provides a distributed and flexible approach to work with different scenarios of IoT. In addition, we have proposed an XACML model for an Adaptive Risk-Based Access Control (AdRBAC) for the IoT and have shown how the access decision will be made using XACML

ACKNOWLEDGEMENTS

We acknowledge Egyptian cultural affairs and mission sector and Menoufia University for their scholarship to Hany Atlam that allows the research to be funded and undertaken.

REFERENCES

Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter., M., 2003. Enterprise Privacy Authorization Language (EPAL 1.2).

Atlam, H.F., Alenezi, A., Alharthi, A., Walters, R., Wills, G., 2017a. Integration of cloud computing with internet of things: challenges and open issues. In: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 670–675.

Atlam, H.F., Alenezi, A., Hussein, R.K., Wills, G.B., 2018. Validation of an Adaptive Risk-based Access Control Model for the Internet of Things. I.J.

Comput. Netw. Inf. Secur. 26–35.

Atlam, H.F., Alenezi, A., Walters, R.J., Wills, G.B., 2017b. An Overview of Risk Estimation Techniques in Risk-based Access Control for the Internet of Things. In: Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017). pp. 254–260.

Atlam, H.F., Alenezi, A., Walters, R.J., Wills, G.B., Daniel, J., 2017c. Developing an adaptive Risk-based access control model for the Internet of Things. In: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 655–661.

Atlam, H.F., Attiya, G., El-Fishawy, N., 2013. Comparative Study on CBIR based on Color Feature. Int. J. Comput. Appl. 78, 975–8887.

Atlam, H.F., Attiya, G., El-Fishawy, N., 2017d. Integration of Color and Texture Features in CBIR System. Int. J. Comput. Appl. 164, 23–28.

Bauer, L., Ligatti, J., David Walker, 2004. A language and system for composing security policies.

Bijon, K.Z., Krishnan, R., Sandhu, R., 2013. A framework for risk-aware role based access control. 2013 IEEE Conf. Commun. Netw. Secur. 462–469.

Bugiel, S., Heuser, S., Sadeghi, A.-R., 2013. Flexible and fine-grained mandatory access control on Android for diverse security and privacy policies. Proc. 22nd USENIX Secur. Symp. 131–146.

Gasparini, L., 2013. Risk-Aware Access Control and XACML.

Hada, S., Michiharu Kudo, 2000. Xml access control language: Provisional authorization for xml documents.

Hu, V.C.V., Ferraiolo, D.F., Kuhn, D.R., 2006. Assessment of access control systems. Nistir 7316 60.

Hulsebosch, R.J., Salden, A.H., Bargh, M.S., Ebben, P.W.G., Reitsma, J., 2005. Context sensitive access control. In: Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies. pp. 111–119.

Janak, J., Nam, H., Schulzrinne, H., 2012. On Access Control in the Internet of Things. Lix.Polytechnique.Fr 1–3.

Jayasekara, A., 2011. Understanding XACML policy langauge. Int. J. Inf. Technol. 35–68.

Jin, X., Krishnan, R., Sandhu, R.S., 2012. A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC. DBSec 12, 41–55.

Kumar, A., Karnik, N.M., Chafle, G., 2002. Context sensitivity in role-based access control. Oper. Syst. Rev. 36, 53–66.

Kumaraguru, P., Cranor, L., Lobo, J., Calo, S., 2007. A Survey of Privacy Policy Languages. In: Security, N SOUPS Workshop on Usable IT Management. pp. 35–40.

Liang Chen, Luca Gasparini, and T.J.N., 2013. XACML and risk-aware access control. In: Proc. ICEIS,. pp. 66–75.

Liu, C., Peng, Z., Wu, L., 2016. Role of Time-Domain Based Access Control Model 57–62.

Liu, J., Xiao, Y., Chen, C.L.P., 2012. Authentication and access control in the Internet of things. Proc. - 32nd IEEE Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2012 588–592.

Molloy, I., Dickens, L., Morisset, C., Cheng, P., Lobo, J., Russo, A., 2011. IBM Research Report Risk-Based Access Control Decisions under Uncertainty 25121.

N.Mahalle, P., Anggorojati, B., Prasad, N.R., Prasad, R., 2013. Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things. J. Cyber Secur. Mobil. 1, 309–348.

Oasis, 2005. eXtensible Access Control Markup Language. OASIS Stand. 141.

OASIS, 2003. eXtensible Access Control Markup Language (XACLML) 1–154.

Rissanen, E., 2010. Xacml v3.0 privacy policy pro le version 1.0.

Suhendra, V., 2011. A Survey on Access Control Deployment. Commun. Comput. Inf. Sci. 11–20.

Turner, K.J., Stephan Rei -Marganiec Lynne Blair, G.A.C.F.W.K.J.T.S.R.L.B.A., Cambpell, G., Wang., F., 2014. Appel: Adaptable and programmable policy environment and language.

VimercatiS, S.D.C. di, S, F., S, J., P., S., 2007. Access Control Policies and Languages in Open Environments. In: Advances in Information Security. Springer, Boston, MA, pp. 21–58.

W3C, 2006. Platform for privacy preferences (P3P) project [WWW Document]. http//www.w3.org/ P3P.

Westphall, C.M., Schmitt, G.R., 2016. A Risk Calculus Extension to the XACML Language. Brazilian Symp. Inf. Syst. 321–328.

Zhou, L., Varadharajan, V., Hitchens, M., 2014a. Secure administration of cryptographic role-based access control for large-scale cloud storage systems. J. Comput. Syst. Sci. 80, 1518–1533.

Zhou, L., Varadharajan, V., Hitchens, M., 2014b. Secure administration of cryptographic role-based access control for large-scale cloud storage systems. J. Comput. Syst. Sci. 80, 1518–1533.

Zhu, H., Jin, R., 2007. A Practical Mandatory Access Control Model for XML Databases A Practical Mandatory Access Control Model for XML Databases. In: 2nd International Conference: Scalable Information Systems. pp. 1–4.

Zhu, Y., Ahn, G.-J., Hu, H., Wang, H., 2010a. Cryptographic Role-based Security Mechanisms Based on Role-Key Hierarchy. Proc. 5th ACM Symp. Information, Comput. Commun. Secur. - ASIACCS '10 314.

Zhu, Y., Ahn, G.-J., Hu, H., Wang, H., 2010b. Cryptographic Role-based Security Mechanisms Based on Role-Key Hierarchy. Proc. 5th ACM Symp. Information, Comput. Commun. Secur. - ASIACCS '10 314.