J Supercomput (2014) 67:711–722
DOI 10.1007/s11227-013-0980-1

Ontology-based access control model for security policy reasoning in cloud computing

Chang Choi · Junho Choi · Pankoo Kim

Published online: 19 July 2013

Abstract There are many security issues in cloud computing service environments, including virtualization, distributed big-data processing, serviceability, traffic management, application security, access control, authentication, and cryptography, among others. In particular, data access using various resources requires an authentication and access control model for integrated management and control in cloud computing environments. Cloud computing services are differentiated according to security policies because of differences in the permitted access right between service providers and users. RBAC (Role-based access control) and C-RBAC (Context-aware RBAC) models do not suggest effective and practical solutions for managers and users based on dynamic access control methods, suggesting a need for a new model of dynamic access control that can address the limitations of cloud computing characteristics. This paper proposes Onto-ACM (ontology-based access control model), a semantic analysis model that can address the difference in the permitted access control between service providers and users. The proposed model is a model of intelligent context-aware access for proactively applying the access level of resource access based on ontology reasoning and semantic analysis method.

Keywords Access control model · Cloud computing · Ontology reasoning · Semantic analysis model

C. Choi · J. Choi · P. Kim (✉)
Department of Computer Engineering, Chosun University, Gwangju, Korea
e-mail: pkkim@chosun.ac.kr

C. Choi
e-mail: enduranceaura@gmail.com

J. Choi
e-mail: xdman@paran.com

1 Introduction

There are many security issues in cloud computing service environments, including virtualization, distributed big-data processing, serviceability, traffic management, application security, access control, authentication, and cryptography, among others. In particular, data access using various resources requires a user authentication and access control model for integrated management and control in cloud computing environments [1, 20, 21].

Cloud computing services are differentiated according to the security policy component [2, 25] because there are differences in the permitted access right between service providers and users. For example, the subject of IaaS (infrastructure as a service) is the system and network manager. Here IaaS can be divided into user accounts, network resources, and system resources, among others. Subjects and objects classified according to services such as IaaS, PaaS, and SaaS now use DAC (discretionary access control), RBAC (role-based access control), and ABAC (attribute-based access control) in cloud computing environments.

In particular, the access control model is the most frequently used method for detecting and preventing insider intrusions [2, 6, 22]. In general, systems for detecting and preventing insider intrusions use RBAC and C-RBAC (context-aware RBAC) models [4, 14]. However, RBAC cannot provide dynamic access control because context-aware elements are not included. C-RBAC does not ensure privacy protection and integrity because it does not consider the level of security between objects. In addition, C-RBAC cannot prevent information leaks using legitimate access method. The recently proposed delegation model does not provide effective and practical solutions to security problems such as information leaks. Therefore, there is a need for a new model of dynamic access control to address the limitations of existing methods based on cloud computing characteristics [3–5, 10].

This paper proposes an ontology-based access control model (Onto-ACM) for dynamic access control. Onto-ACM is a semantic analysis model [11, 12, 23, 24] that can address differences in the permitted limit between service providers and users. The proposed method[1] is an intelligent and context-aware access model for proactively applying the security level of resource access based on ontology reasoning and semantic analysis method.

2 Related work

2.1 Role-based access control model

The RBAC model [7–10] is versatile and conforms closely to the organizational model used in firms. RBAC meets this requirement by separating users from roles. Access rights are given to roles, and roles are assigned to users. Here the role combines users and privileges [13]. The basic concepts of RBAC originated with early multi-user computer systems. The resurgence of interest in RBAC has been driven

[1] This paper extends our previous work published on MIST 2012 [18].

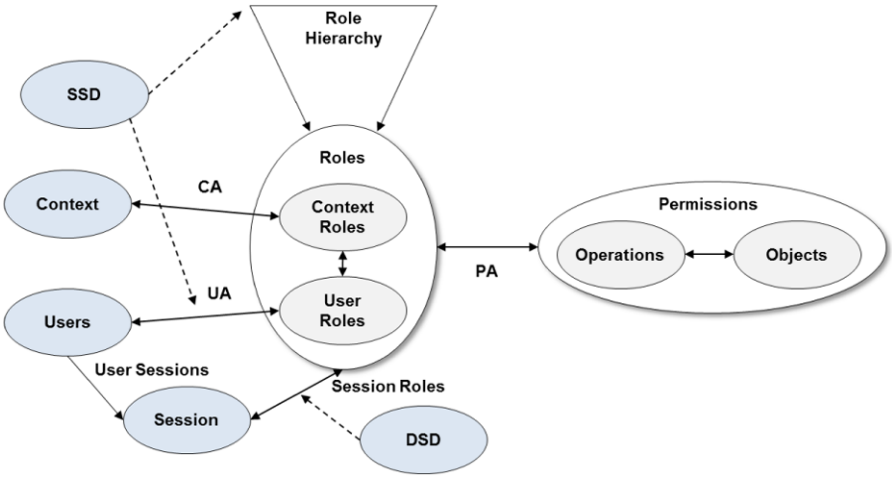

Fig. 1 C-RBAC Model[14]

by the need for general-purpose customizable facilities for RBAC and the need to manage the administration of RBAC itself. As a consequence RBAC facilities range from simple to complex [16].

2.2 Context-aware role-based access control model

C-RBAC is an extension of traditional role-based access control model that allows security administrators to define context oriented access control policies enriched with the notion of purposes. By adding C-RBAC roles, they extend traditional access control model that helps organizations to know which user can perform what operation on which object with what purpose [14]. Zou [17] proposed imposing multi-grained constraints on the RBAC model in the multi-application environment and it shows the authorization process of the proposed model.

Figure 1 is an extended C-RBAC model and access control model for security context information based on context-role in ubiquitous computing system. This model is given the additional features of role active/inactive, hierarchy role and so on.

2.3 Context aware-task role based access control

CA-TRBAC (Context aware-task role based access control) is a control access and prevent illegal access efficiently for various information systems in ubiquitous computing environment. CAT-RACS (Context-aware task-role based access control system) applied CA-TRBAC, which adds context-role concept tor achieve policy composition by context information and security level attribute to be kept confidentiality of information. It provides security services of user authentication and access control by context-aware security manager, and provides context-aware security services [15].

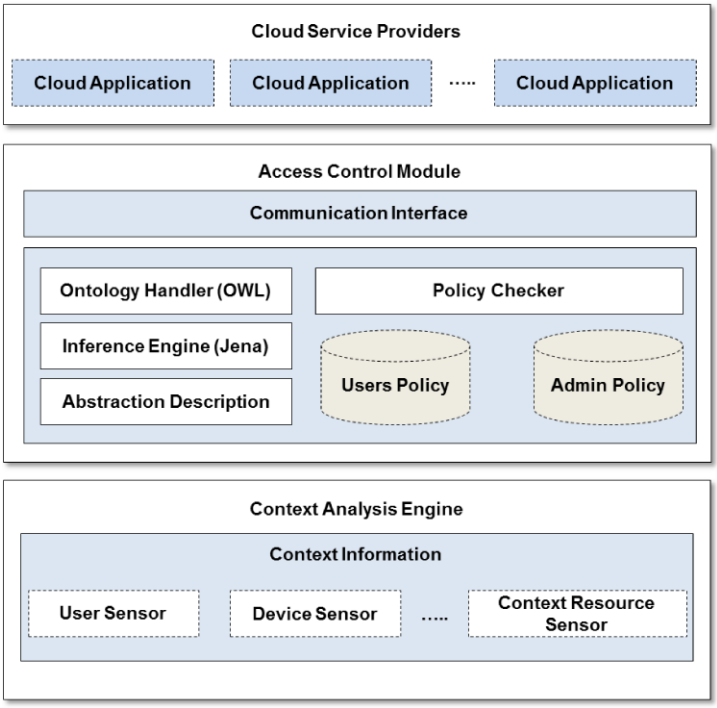

Fig. 2 Onto-ACM framework

3 Access control model based on ontology in the cloud computing environment

3.1 Proposed frameworks

Detailed and dynamic access control in cloud computing environments entails several requirements:

1. The user's role can be dynamically and partially delegated by changing permission.
2. Constraints on the authorized role can be considered for dynamic access control.
3. Objects, conditions, and obligations for data access can be considered for protecting information in the database.
4. Data access as necessary can be rejected.
5. Access control can be given based on location and equipment needs.
6. The most important factor is the prevention of any misuse of access rights.

Onto-ACM offers a mechanism for securing applications and systems considering the above conditions based on context-aware technologies in the cloud computing environment.

Figure 2 shows the Onto-ACM system architecture. Onto-ACM consists of the context-aware security manager for context-aware security services; the context analysis engine for the selection, analysis, integration, and provision of context-aware

information; and the access control module for the composition and management function of the security policy based on the context-aware communication interface for the composition and management interface of the security policy.

The access control module requires some security policy and context information for a user's authentication and access control based on the user's request for access from the context analysis engine. The access control module provides a security policy related access control and context information. Finally, the ontology reasoning process is based on the integration of context information in the access control module.

The context analysis engine permits system access through the conformity of the security policy and the context condition. The ontology handler provides the location of all resources that can be accessed based on role and context information. This method limits access to resources through the access policy in the cloud computing environment.

① ***Context analysis engine***
The context analysis engine manages the gathering and management of context information for security services based on context-aware information in the cloud computing environment. In addition, this engine sends queries to the access control module through query creation for information gathering.

② ***Access control module***
The access control module manages security services such as user authorization, access control, and context information in the cloud computing environment. Onto-ACM provides security services such as user identification, authorization, and access control for the use of applications through the context analysis engine. In addition, this module provides security services for authorization and access control based on the provision of context information from the context analysis engine.

③ ***Inference engine***
The access control inference engine performs the access control function in the proposed context-aware access control system. The inference engine consists of authorization services, permission services, and context information ontology, among others. Each module manages the security policy and controls access rights by information resources through inferences based on the role of the active user and that of the context.

④ ***Ontology handler***
The ontology handler manages the ontology of context information through data processing for the context information repository and authorization services based on user access. Context information ontology includes a transaction list for the access demand and information on approval rules for the permission of each transaction and uses OWL (ontology web language) for gathering and analyzing context information. The inference engine performs the reasoning of the access control policy.

⑤ ***Policy checker***
The policy checker performs the subject's identification, management, and processing of context information and provides dynamically allocated services based

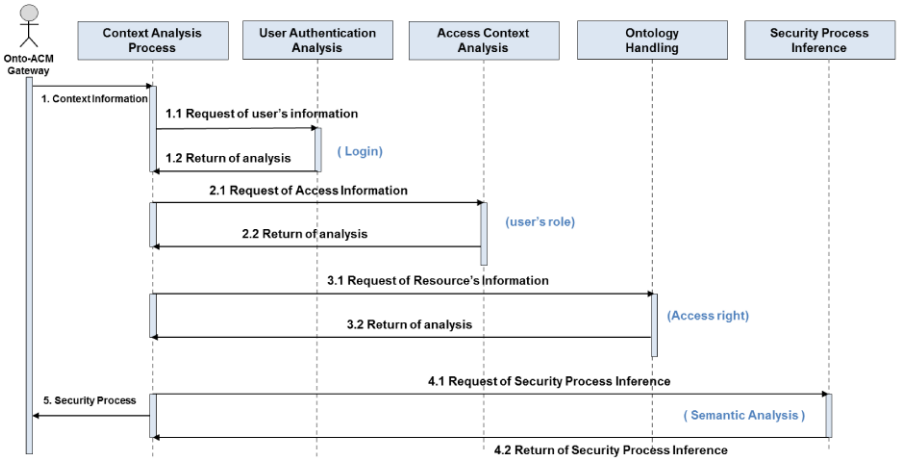

Fig. 3 Sequence diagram of Onto-ACM

on the user's role by obtaining additional information and analyzing access policy based on the access location and time and the spatial area. Finally, the policy checker makes access control decisions through a comparative analysis of the current user's active role, currently active situations, and the security policy.

⑥ ***Abstraction description***
The abstraction description monitors access control for the administrator and users and provides context information reasoning for semantic analysis from the sensor and equipment.

3.2 Context-aware security system

Figure 3 shows a sequence diagram of the context-aware security system. The Onto-ACM process is as follows:

① The internal user accesses the access control module for the certification processing of resource access. The context analysis engine gathers context information on the user for the internal authorization of the user based on cloud resources. The access control module analyzes gathered information on the user context.

② The access control module connects the context information ontology repository for the user's role assignment and access control.

③ The access control module is required for information on the user's role and access policy data from the context information ontology repository. The policy checker of the access control module grants access rights through the user's role and the access policy for resource access.

④ An internal user requires services through the acquisition of access rights. The access control module accesses appropriate resources through access level required resources. Context information ontology is used for making decisions on appropriate resources through inferences based on context information on the user, the security policy, and access control.

Table 1 Example of the Onto-ACM policy

[System Policy]
Policy : AdminPolicy, DataOwnerPolicy
Permission 1, Permission 2 Permission
Permission : Accept, Reject
[Admin Policy]
AdminPolicy : Permission 1 Role Action To Access
[DataOwner Policy]
DataOwnerPolicy : Permission 2 Access Action To Context Data

Table 2 Classification of context information

Information	Example of context information
Identity context	User rights
Physical context	User location, terminal, and security status
Preference context	System and resource access time
Behavioral pattern context	Frequency of access and main commands
Resource context	Resource access rights

4 Context-aware ontology based access control policy

4.1 Definition of the Onto-ACM policy

The owner has to know all types of services and have information on each service in the cloud computing environment when existing access control methods are used to make an access policy because access control methods can facilitate policymaking by the administrator and the owner. The administrator has to know each user's information and access status in the cloud computing environment when existing access control methods make an access policy by the administrator. Therefore, existing access control methods can be difficult to apply because of large-scale systems and large numbers of users. This paper divides Onto-ACM into the user policy and the administrator policy. The administrator describes the service policy by special roles. In addition, the owner describes a user-defined policy with the level of information access for a special object. In Table 1, the role is the particular position and function of the user or service in the system. The advantage of this role is the description of an efficient policy because the user or administrator does not directly describe the policy. Even when the administrator policy permits special services, there cannot be access to special information by this role if the user does not permit such services.

4.2 Context information class

In this paper, the context information class consists of identity, physical, preference, and other information. Table 2 shows an example of context information.

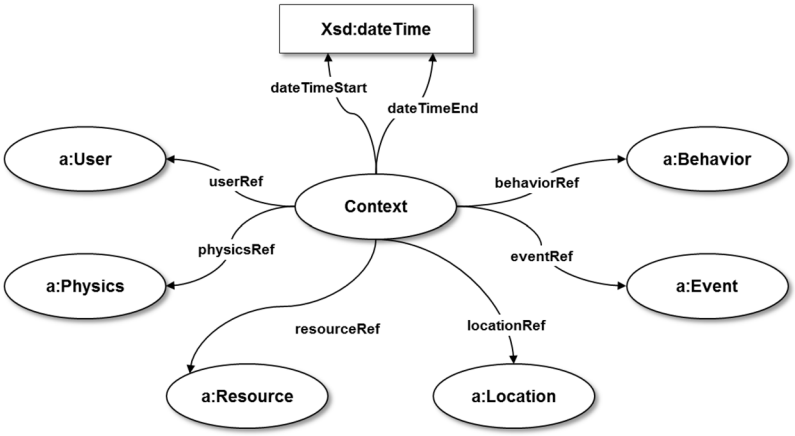

Fig. 4 Classes and properties of context ontology

Onto-ACM defines the context information ontology of the user and the administrator by using OWL based on the ontology class, including basic information, the resource time, and the terminal, among others. Figure 4 classifies context information based on context ontology classes and properties. Table 3 shows the OWL source code for Onto-ACM context ontology.

Table 3 shows the sample OWL code for Onto-ACM context ontology.

5 Applying and comparative evaluation of the Onto-ACM

The authentication and access control processing in security system is tested for an experiment of an access control model based on context reasoning in cloud computing environments. The data access using various resources requires an authentication and access control model for integrated management and control in cloud computing environments. Cloud computing services are differentiated according to security policies because of differences in the permitted access right between service providers and users. Therefore, the information management layer controls authentication and access right through ontology reasoning of conditions. Table 4 shows an example of administrator and user policies based on ontology reasoning.

Figure 5 is a processing of context ontology inference by a semantic reasoner. The user is connected to the main system using login process. And then, the user is authorized an access right of resources.

Figure 6 shows creation processing of access right through authentication using an ontology reasoning engine (Jena [19]).

The user can confirm information of administrator and user through creation processing of access right automatically. The access right of the proposed model consists of shared resources and private resources.

This paper proposes an access control model using context reasoning such as context, permission level, condition on permission, purpose, and each policy for administrator and user. Also, we analyze the requirements of access control model using

Table 3 Sample OWL code in Onto-ACM context ontology

```
<owl:Class rdf:ID="CloudContext">
        <rdfs:hasPhysics rdf:resource="#Physics"/>
        <rdfs:hasUser rdf:resource="#User"/>
        <rdfs:hasEvent rdf:resource="#Event"/>
        <rdfs:hasResource rdf:resource="#Resource"/>
</owl:Class>

<owl:Class rdf:ID="User">
        <rdfs:subClassOf rdf:resource="#CloudContext"/>
</owl:Class>

        <owl:Class rdf:ID="Admin">
                <rdfs:subClassOf rdf:resource="#User"/>

                <rdfs:hasHistory rdf:resource="#History"/>
<rdfs:hasHistory rdf:resource="#Location"/>
<rdfs:hasHistory rdf:resource="#Terminal"/>
                <rdfs:hasDepartment rdf:resource="#Department"/>
                <rdfs:hasOption rdf:resource="#Option"/>
                <rdfs:hasAdmin-Resource rdf:resource="#Admin-Resource"/>
                <rdfs:hasAdmin-Event rdf:resource="#Admin-Event"/>
        </owl:Class>

                        <owl:Class rdf:ID="Admin-Resource">
                                <rdfs:subClassOf rdf:resource="#Admin"/>
                        </owl:Class>

                                <owl:ObjectProperty rdf:ID="System">
                                        <rdfs:domain rdf:resource="#Admin-Resource"/>
                                        <rdfs:range rdf:resource="#Resource"/>
                                </owl:ObjectProperty>

                        <owl:Class rdf:ID="Admin-Event">
                                <rdfs:subClassOf rdf:resource="#Admin"/>
                        </owl:Class>

                                <owl:ObjectProperty rdf:ID="User-Create">
                                        <rdfs:domain rdf:resource="#Admin-Event"/>
                                        <rdfs:range rdf:resource="#Event"/>
                                </owl:ObjectProperty>
```

Table 4 Example of Administrator and user policies based on ontology reasoning

Policy P1 = subject='Admin1', rules=R1, R2, R3
Rule R1 = 'read', 'permit', Class(User)
Rule R2 = 'read', 'permit'Class(CloudCotext)
Rule R3 = 'read', 'deny', Individual(Admin-Resource)
Policy P2 = subject='user1', rules=R3, R4, R5
Rule R3 = 'read', 'permit', Class(User)
Rule R4 = 'read', 'deny', Individual(DiskCopy)
Rule R5 = 'read', 'permit', DatatypeProperty(hasAllocDisk)

cloud computing characteristics. Table 5 is shows the sample of ontology reasoning results.

In the cloud computing environment, the user is given authorization information through inferences based on context ontology. In Onto-ACM, inference processing is based on the Jena inference engine, and query processing, on SPARQL. Table 6 shows the results for inferences based on Onto-ACM context ontology.

In Table 6, context information is obtained through the user ID, the resource IP, the client IP, client information, and the access time, among others. Even when Onto-

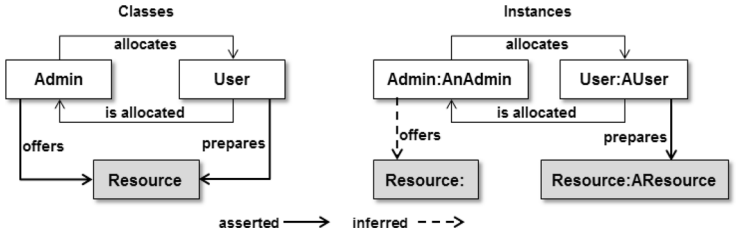

Fig. 5 Processing of context ontology inference by semantic reasoner

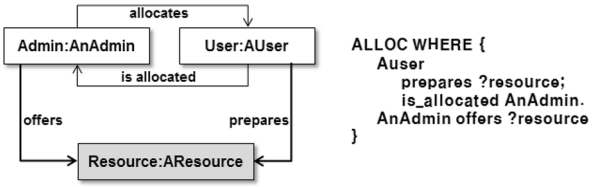

Fig. 6 Resource allocation using SPARQL based on context ontology

Table 5 Classification of Context Information

	Task 1	Task 2	Task 3
Permission	Permission 1	Permission 2	Permission 3
Service	Sales Management (User 1)	Customer Management (User 2)	Performance Management (Administrator)
Context	Device	Place, Device	Place
Context Resource	Office PC	Office PC, 12:00-16:00	Main Server (Security System)
condition on permission	Share	Private	Share
Permission Level	Weak	Weak	Strong
Admin Approval	N	N	Y
Log Information	N	Y	Y

ACM involves the same user and situation, the procedure for addressing security varies according to the access time and the user location.

The advantage of the proposed model is in convenience and efficiency of policy management. The delegation of C-RBAC model just grants a role inheritance by administrator but proposed model can grant a role inheritance by administrator and user. Therefore, it is useful to protect malicious information leakage. Also, proposed model can be a detailed dynamic access control that can address the limitations of cloud computing characteristics. Namely, Onto-ACM provides inference for access control decision making, and allows access control information to be searched, queried and discovered automatically. Finally, Onto-ACM has added a considerable generality to

Table 6 Example of administrator and user policies based on ontology reasoning

[Result 1]
User ID: “Admin” [User Permissions: admin. History: everyday]
Client IP: “203.237.97.11”[Network: LAN, Location: office]
Client Info: “PC”, “High” [Device: PC, Battery Status: High]
Resource IP: “117.16.23.170” [Access Level: Low]
Access Time: “23::12:11” [Access Time: working]
[Result 2]
User ID: “user1” [User Permissions: user, History: everyday]
Client IP: “202.217.112.15”[Network:VPN, Location: Home]
Client Info: “PC”, “High” [Device: PC, Battery Status: High]
Resource IP: “.117.16.23.177” [Access Level: High]
Access Time: “02:24:32”[Access Time: off]

the model, for example the context ontology are going to be universally used for user authentication and authorization.

6 Conclusion

Access control models represent the most frequently used method for detecting and preventing insider intrusions. In general, systems for detecting and preventing insider intrusions are based on RBAC and C-RBAC models. However, RBAC cannot provide dynamic access control because it includes no context-aware elements. C-RBAC does not ensure the protection of privacy and integrity because it does not consider the level of security in between. In addition, C-RBAC (Context-aware RBAC) models do not suggest effective and practical solutions for managers and users based on dynamic access control methods, suggesting a need for a new model of dynamic access control that can address the limitations of cloud computing characteristics.

This paper proposes an access control model using context reasoning such as, context, permission level, condition on permission, purpose, and each policy for administrator and user. Also, we analyze the requirements of the access control model using cloud computing characteristics. In the cloud computing environment, the user is given authorization information through inferences based on context ontology. In Onto-ACM, inference processing is based on the Jena inference engine, and query processing, on SPARQL.

The advantage of the proposed model is in convenience and efficiency of policy management. The delegation of C-RBAC model just grants a role inheritance by administrator but the proposed model can grant a role inheritance by administrator and user. Therefore, it is useful to protect malicious information leakage. Also, the proposed model can be a detailed dynamic access control that can address the limitations of cloud computing characteristics.

Acknowledgements This study was supported by research fund from Chosun University, 2012.

References

1. Li X, He J (2011) A user-centric method for data privacy protection in cloud computing. In: 2011 international conference on computer, electrical, and systems sciences and engineering, pp 355–358
2. Bowen BM, Ben Salem M, Hershkop S (2009) Designing host and network sensors to mitigate the insider threat. IEEE Security Privacy Mag 7(6):22–29
3. Ferraiolo DF, Richard Kuhn D, Chandramouli R (2003) Role-based access control. Artech House, Norwood
4. Corradi A, Montanari R, Tibaldi D (2004) Context-based access control for ubiquitous service provisioning. In: Proceedings of the 28th annual international computer software and applications conference, Sep. IEEE Press, New York, pp 444–451
5. Han W, Zhang J, Yao X (2005) Context-sensitive access control model and implementation. In: Proceedings of the fifth international conference on computer and information technology. IEEE Press, New York, pp 757–763
6. Cappelli D, Moore A, Trzeciak R, Shimeall TJ (2006) Common sense guide to prevention and detection of insider threats. Carnegie Mellon University
7. Ahn G-J, Sandhu R (2000) Role-based authorization constraints specification. ACM Trans Inf Syst Secur 3(4):207–226
8. Bertino E, Bonatti PA, Ferrari E (2001) Trbac: a temporal role-based access control model. ACM Trans Inf Syst Secur 4(3):191–233
9. Joshi JBD, Bertino E, Latif U, Ghafoor A (2005) A generalized temporal role-based access control model. IEEE Trans Knowl Data Eng 17(1):4–23
10. Li N, Tripunitara MV (2006) Security analysis in role-based access control. ACM Trans Inf Syst Secur 9(4):391–420
11. Finin T, Joshi A, Kagal L, Niu J, Sandhu R, Winsborough W, Thuraisingham B (2008) ROWLBAC: representing role based access control in OWL. In: Proceedings of the 13th ACM symposium on access control models and technologies. ACM, New York, pp 73–82
12. Macfie A, Kataria P, Koay N, Dagdeviren H, Juric R, Madani K (2008) Ontology based access control derived from dynamic RBAC and its context constraints. In: Proceedings of the 11th international conference on integrated design and process technology (IDPT 2008), Taichung, Taiwan, 1–6 June 2008
13. Kalajainen T (2007) An access control model in a semantic data structure: case process modelling of a bleaching line. Department of Computer Science and Engineering
14. Nabeel Tahir M (2007) C-RBAC: Contextual role-based access control model. Ubiquitous Comput Commun J 2(3):67–74
15. Eom J-h, Park S-H, Chung T-M (2008) A study on architecture of access control system with enforced security control for ubiquitous computing environment. J Korean Inst Inf Secur Cryptol 18(5):71–81
16. Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 29(2):38–47
17. Zoua D, Heb L, Jina H, Chenc X (2009) CRBAC: imposing multi-grained constraints on the RBAC model in the multi-application environment. J Netw Comput Appl 32(2):402–411
18. Choi C, Choi J, Ko B, Oh K, Kim P (2012) A design of onto-ACM(Ontology based access control model) in cloud computing environments. J Internet Serv Inf Secur 2(3/4):54–64
19. Apache Jena Project (2013). http://jena.apache.org/
20. Kiyomoto S, Fukushima K, Miyake Y (2011) Towards secure cloud computing architecture—a solution based on software protection mechanism. J Internet Serv Inf Secur 1(1):4–17
21. Pieters W (2011) Representing humans in system security models: an actor-network approach. J Wirel Mobile Netw Ubiquitous Comput Depend Appl 2(1):75–92
22. Zia TA, Zomaya AY (2011) A lightweight security framework for wireless sensor networks. J Wirel Mobile Netw Ubiquitous Comput Depend Appl 2(3):53–73
23. Jung JJ (2012) Evolutionary approach for semantic-based query sampling in large-scale information sources. Inf Sci 182(1):30–39
24. Jung JJ (2012) ContextGrid: a contextual mashup-based collaborative browsing system. Inf Syst Front 14(4):953–961
25. Jung JJ (2011) Service chain-based business alliance formation in service-oriented architecture. Expert Syst Appl 38(3):2206–2211