International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2019/187283 Volume 6, Issue 5, September – October (2019)

RESEARCH ARTICLE

Two-Level Grid Access Control Model Based on Resource Performance and Request Priority

Sarra Namane
Networks and Systems Laboratory, Badji Mokhtar University, Annaba, Algeria.
naamanesara2005@yahoo.fr

Nacira Ghoualmi
Networks and Systems Laboratory, Badji Mokhtar University, Annaba, Algeria.
ghoualmi@yahoo.fr

Published online: 29 October 2019

Abstract – This paper summarizes all the mechanisms presented previously to efficiently represent grid computing security policies. The prime advantage of this work is to reduce the complexity by minimizing the number of security rules that require checking. This is achieved by combining the Grid Authorization Graph (GAG) and the Weighted Grid Authorization Graph (WGAG) in a two-level access control model. This model attributes the grid resources to users according to the performance that the job requires, thus avoiding performance waste. Simulations results showed the effectiveness of the proposed model in reducing the number of security rules that require checking.

Index Terms – Authorization, Access Control, BFA, HCM, PCM, GAG, WGAG, Resources Performance, Grid Security.

1. INTRODUCTION

Recently, grids are considered as a technology of great potential in both academia and industry [1]. They are used by many organizations to resolve high computing problems. Varied concerns about grid security have been arising, appropriate authorization and access control is a key problem faced by this new paradigm [2]. The access control process can be divided into two steps [3]. The first step concerns the security policies storing and management. While the second takes into consideration the access control itself (how to answer an access control request rapidly and efficiently) [4, 5, 6]. In this last step, the system verifies resources' security policies in order to make the access decision. This process takes a lot of time because it requires a rule by rule checking in each resource security policy. Several studies have presented effective mechanisms for resources security policies storing. In [7], the authors tried to ensure a quick response to users' access requests by presenting a well-defined data structure. The Primitive Clustering Mechanism (PCM) [7], the Hierarchical Clustering Mechanism (HCM) [8, 9, 10] and the Grid Authorization Graph (GAG) [11] are presented to decrease the redundancy and repetition in security policy verification using a tree structure to arrange resources security policies. However, parsing the entire graph is necessary in case of access request for one resource or for multiples resources. Each resource in grid environment has its own properties [12] that might be similar or somewhat identical to other properties relevant to other resources. Due to this fact, this study gathers resources that have similar performance in the same group, then a meta-authorization system is introduced to present the meta-security policy of each group in the form of a Weighted Grid Authorization Graph (WGAG) that manages the two aspects: resource performance and a cross domain access control process in grid computing environments. The proposed model combines the Grid Authorization Graph (GAG) and the Weighted Grid Authorization Graph (WGAG) in two different levels to minimize how many security rules should be verified.

The remainder of this article is structured as: section 2 presents the recently published literature related to access control models. Section 3 gives a description of the suggested architecture in grids. In section 4, the model was simulated in order to evaluate its effectiveness. Section 5 gives a discussion about the advantages of the proposed model. At last, section 6 is a conclusion that includes the prospective future work.

2. RELATED WORKS

In [11], the authors summarized the existing manners to represent security policies efficiently in grid computing environment. They started by the Brute Force Approach (BFA), this mechanism needs to verify all security policies to provide the set of resources reachable by a user. Subsequently, the authors noted that some resources have the same security rules. This point allowed them to group these resources together by introducing the Primitive Clustering Mechanism (PCM). Then, the authors perceived that PCM eliminates the redundancy of the same security policies verification but it cannot eliminate the redundancy of the identical security rules verification. The authors proposed the Hierarchical Clustering Mechanism (HCM) to shun the

RESEARCH ARTICLE

redundancy found in PCM. For this purpose, the authors used parent node's information like data in order to produce a hierarchical clustering of the parent nodes based on their common security rules. Then, the authors analyzed the HCM mechanism, they found that it effectively reduces the redundancy during the verification of the security rules but unfortunately this redundancy still exists. Furthermore, HCM does not permit of the OR-based security policies representation. To deal with these negative points, the authors proposed a Grid Authorization Graph (GAG). They extended the HCM tree to a graph by introducing a correspondence edge which allows for security rules checking without any redundancy.

In [13], the authors noticed that the Grid Authorization Graph (GAG) completely removes the repetition in security rules checking. Nevertheless, some security rules did not require verification from the beginning and yet the GAG checked them. This point allowed the authors to avoid once again other security rules verification. They focused on the mathematical knowledge of graph theory [14]. Inspired by the Bell Lapadula model [15], they assigned a weight to every edge in the GAG. Then, they associated a classification level to every resource in the graph. This attribute represents the shortest path between the resource node and the root node. Besides, the authors gave a security clearance attribute to each user in the system. This attribute is derived from user set of roles. The main principle of the WGAG is to eliminate parsing the graph for resources that have a classification level higher than user security clearance. In this way, the authors were able to eliminate the redundancy of checking several security rules from the beginning.

In [16], the authors demonstrated the inefficiency of the HCM mechanism when the authorization request intends a specific resource. HCM wastes the time and complicates the authorization process when it parses the whole decision tree while it is sufficient to check a single resource security rule. The authors presented a hybrid authorization mechanism to overcome this problem. They took into consideration the cross domain authorization aspect by using a model that is composed of central HCM and HCM agent in each domain. Each agent builds a local security policies tree using the Counting Algorithm [8]. At each received access control request, the central HCM must propagate the request to its registered agents across the grid domains. Every local HCM Agent parses its decision tree and gives as a response to the Central HCM a group of resources that the user can access. The Central HCM concatenates the different sets to acquire the whole set of resources that the user can to access. The proposed model has the advantage of fast parsing of the decision tree as it is distributed across several domains. However, it has the disadvantage of network latency due to the communication required between the central HCM and its agents for each authorization request.

2.1. The limitations of the existing approaches

All the mechanisms presented previously attempted to reduce the redundancy of security rules verifications during the access control process. However, none of them takes into consideration the reduction of the graph size criterion. Reducing this size is a very important criterion because it implies the reduction of the complexity and as a consequence the reduction of the answer time to an access control request. Moreover, grid computing resources have different performances; thus, attributing the adequate resource (according to its performance) to the correspondent job permits to avoid performance waste and ensures a better resource use. This last criterion was not taken into account by all the works proposed in the literature.

3. PROPOSED MODELLING

Grids are generally organized in the form of multiple administrative domains. Each domain has its own set of resources and its users [17]. Each resource in grid computing environment has its own properties such as: a performance level that might be similar or somewhat identical to other performance levels relevant to other resources. This study gathers the resources that have similar performance levels in the same group, where each group has its own meta-security policy which is a set of Meta-roles. All meta-security policies are represented in the form of a weighted tree where vertices are Meta- roles, leaves are groups and the weight of each edge represents the importance degree of the Meta-role wherefrom the edge comes out. A “Performance Level (PL)” attribute can be given to every group. PL represents the weight’s value relevant to the shortest path connecting the group’s node to the root node. A “Priority Clearance (PC)” attribute is derived out from the jobs’ needs (whether or not this resource has a high performance) and origin of request (same domain or different one) (step one of the proposed algorithm). It is possible to eliminate the decision graph parsing for authorizing a request Req_i whose priority clearance (PC (Req_i)) is fewer than the performance level of group G_j (GPL (G_j)), that is (PC (Req)) < (GPL (G_j)). Thus the developed Meta-Security Policy Weighted Authorization Graph (MWGAG), which is based on the Grid Authorization Graph (GAG) [11] and the Weighted Grid Authorization Graph (WGAG) using the ABAC [18] model [13] with the following details:

- Consider Groups= $\{G_i|i=1\ldots k\}$ as the collection of grid groups.
- Consider G_i= { rj | j=1 … l } as the collection of resources in G_i having same performance level.
- Let Requests = $\{Req_i|i=1\ldots k\}$ be the set of authorization requests.

RESEARCH ARTICLE

- Consider SR= {sr_k | k=1 … m} as the collection of all security rules.
- Let MSR= { Meta-sr_j | j=1 … l } be the set of all Meta-security rules
- For every resource r_j in G_i, there is a correspondent security policy $SP_j \subseteq SR$.
- For each group G_i in the grid there is a correspondent Meta-security policy $MSP_j \subseteq MSR$
- Consider G (N; A) as the Meta-Security Policy Authorization Graph (MPAG). N constitutes the collection of vertices (Meta- roles) and A constitutes the collection of edges, leaves represent groups.
- Consider MRole= {$mrole_i$|i=1…k} as the collection of user's Meta-roles.
- Let IDMRole: MRole → N: IDMRole ($mrole_j$) = the Importance Degree of the Meta-role $mrole_j$.
- Consider W: E→ N: W (e_{ij}) = IDSR ($mrole_j$) as a function that specifies edges' weights in MPAG. Where $mrole_j$ is the Meta-role wherefrom the edge e_{ij} comes out.
- Consider U = {U_i |i=1…k} as the collection of grid users.
- Consider Role = {$role_i$|i=1…s} as the collection of user's roles.
- Consider SP: Groups→ N: SP (G_j) = the weight of the shortest path connecting the root node to the group G_j in MPAG.
- Consider GPL: Groups → N: GPL (G_j) = SP (G_j) as a function that associates each group to its performance level.
- Consider RPC: Requests →N: RPC (Req_i) = $\sum_{Meta\text{-}roles \in MRole\ (Reqi)}$ IDR ($role_j$) as a function that associates each request to its priority clearance.
- Consider RMR C Requests × MRole: as the collection of relations of requests to Meta-role assignements.
- Consider RMROLE: Requests→ MRoles : RMROLE (Req_i) = {meta-$role_j$ | (Req_i, Meta-$role_j$) ϵ RMR} as a function derived from RMR that associates each request to a collection of Meta-roles.
- Let GAG-Graphs = {G_i'(N'; A') | i=1.. k} be the set of Grid Authorization Graphs where N' constitutes the collection of vertices (roles) A constitutes the collection of edges, the leaves represent resources.

The aim of meta-security policy is restricting access to resources based on the user's needs (resource powerful), the user role and the domain from which the authorization request is generated. A two-level authorization process is proposed based on the Bell-Lapadula Model (BLM) [15], also known as the multi-level model [19] which is a subtype of the Mandatory Access Control model (MAC) [20].The multi-level model permits to enforce access control in military applications and governmental uses. In this type of applications, objects and subjects are often divided into several security levels. Access to objects by subjects is done according to priority degree. Moreover, the role of the user in one administrative domain is different from the one that he obtains in another domain. A note is given to the user depending on its needs (powerful resource or not and domain from which the access control request is generated).

Adding meta-security rules allows for avoiding the verification of all security rules of all resources of the system and only checking the group security rules that the user can access. The access control model presented in this article is a cross domain access control [21] using the Grid Authorization Graph (GAG) and Meta-Security Policies Weighted Graph based on Weighted Grid Authorization Graph (WGAG). An overview of the proposed model is showed in Figure (1). The proposition extends the XACML standard [22] by adding the next new components:

- Meta-security policies decision point (MP-PDP)
- The Request Analyzer and Processor (RAP) [11]
- Weighted Grid Authorization Graph Search engine [13]
- The Weighted Grid Authorization Graph database [13]
- The Extensible Markup Language parser. [11]
- The GAG Search engine [11]
- The GAG database [11]

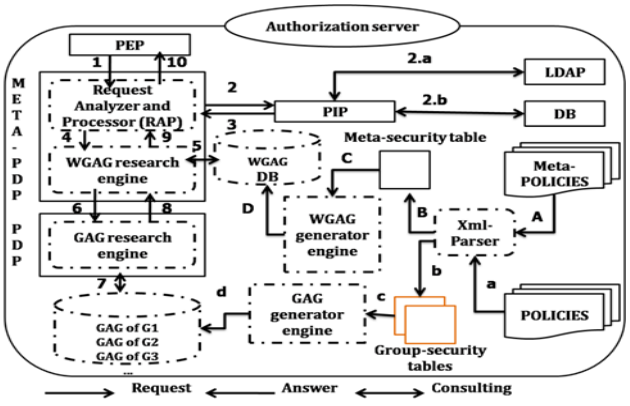

Figure 1: Proposed Architecture

RESEARCH ARTICLE

If a user Ui from domain AD1 wants to access resources from domain AD1 or AD2, he must send its access control request and wait for the access control decision. He will obtain the set of resources that he may access. The access control decision combines the results of two important processes. The first process consists of choosing the groups that the user can access using the nature of resources (powerful, not powerful) and the nature of the request (from the same domain, from a different domain). The second process permits to parse the Grid Authorization Graph (GAG) of the authorized groups to have the set of resources that the user can access (step two of the proposed algorithm).

To better understand the proposed model, the following example is considered: a grid environment with 4 resource groups: Groups = {G1, G2, G3, G4} with four security levels and four Meta-Roles: MRoles= {SameDomain, DifferentDomain, powerful, Notpowerful}, and four Meta-security rules: MSR = {Meta-sr1, Meta-sr2, Meta-sr3, Meta-sr4} with their importance degrees (depicted in table 1).

Meta-sr1 requires the request to be from SameDomain.

Meta-sr2 requires the request to be from DifferentDomain.

Meta-sr3 requires the request to ask for a powerful resource.

Meta-sr4 requires the request to ask for a No powerful resource.

Let G1, G2, G3 and G4 a resource's group as the following:

- G1= {r1, r2, r3, r4}
- G2= {r5, r6, r7, r8}
- G3= {r9, r10}
- G4= {r11, r12}

Let SR= {sr1, sr2, sr3, sr4}

Meta-Security rule	Importance degree
Meta-Sr1	3
Meta-Sr2	2
Meta-Sr3	1
Meta-Sr4	6

Table 1: Meta-Security Rules Importance Degree Table

The 4 resource groups have the following Meta-Security Policies (illustrated in table 2):

- G_1 require Meta-sr_1 and Meta-sr_3.
- G_2 requires Meta-sr_1.
- G_3 requires Meta-sr_1 and Meta- sr_3.
- G_4 require Meta-sr_2 and Meta-sr_4.

Groups \ Meta-sr	Meta-sr_1	Meta-sr_2	Meta-sr_3	Meta-sr_4
G_1	1	0	1	0
G_2	1	0	0	1
G_3	1	0	1	0
G_4	0	1	0	1

Table 2: Meta-Security Table

3.1. Algorithm 1: The Proposed Algorithm

Step one: building of the Weighted Grid Authorization Graph (WGAG) and the Grid Authorization Graphs (in parallel).

A. Meta-security policies of the grid system are submitted by the administrator using the eXtensible Access Control Markup Language (XACML) [22] proposed by the OASIS consortium [23].

B. An XML parser [24] is required to browse the XML files and it gives as a result a Meta-Security Table (table 2).

C. The WGAG generator engine creates a Weighted Grid Authorization Graph (WGAG) using the Meta-Security Table generated by the XML parser. The vertices of the graph represent the Meta-Security Rules, but the weighted edges will have the importance degree of Meta- Security Rules as a value using Meta-security rules importance degrees table (table 1). The leaves of the graph represent the groups. For each Group G_i, the WGAG search engine parses the graph and calculates the shortest path between the root node and G_i. The obtained result is taken as Performance Level of the group G_i.

D. The output weighted decision graph (Figure 2) is maintained in the WGAG database (Figure 1).

a. Each group security policy will be submitted by the administrator to the policy administration point (PAP) per the XACML.

b. The xml parser is used to brose the XML security policies files. A security table for each group (Table 3, 4, 5, 6) is given as a result. These tables will be used as an input to the GAG Generator engine [11].

c. The GAG Generator engine draws the Grid Authorization Graph (GAG) of each group.

d. The output decision graphs are maintained in the GAG database (Figure 1).

RESEARCH ARTICLE

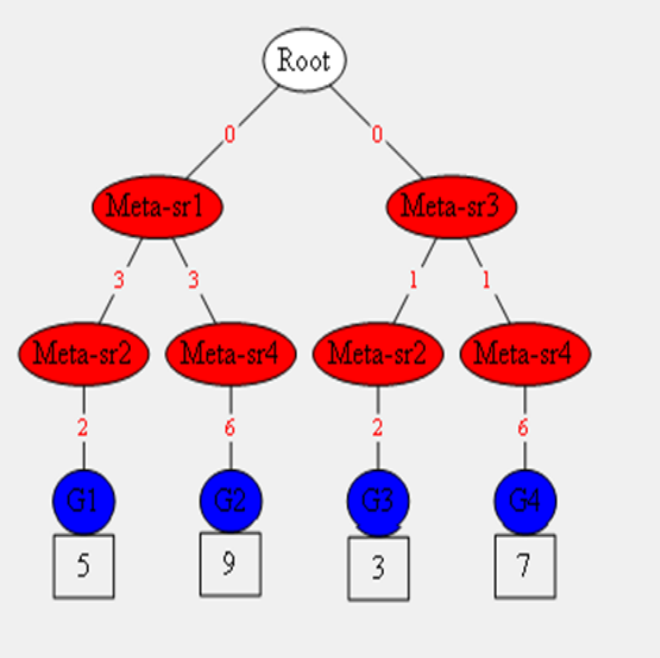

Figure 2: Meta-Security Policy Weighted Tree

Step two: make an access control decision

When an access control request (Req_j), is generated by a user,

1 The PEP takes it and sends it to the Meta-PDP. The request is stored in a queue.

2 The RAP takes the request; send it to the PIP to have attributes values. The PIP consults the LDAP [25] (2.a) and the database (2.b) to have the authorization attributes (User role, request attribute, Meta-security rule importance degree) and calculates the Priority Clearance of the request.

3 The PIP sends the attributes values to the RAP.

4 The Request Analyzer and Processor (RAP) forwards the request to the Weighted Grid Authorization Graph Search Engine [13].

5 The last engine will browse the weighted graph and provide the groups that verify: SL (G_i) ≤ PC (Req_j),

6 The Meta-PDP sends an access request to the PDP with the authorized groups and user roles.

7 The GAG Search engine parses each graph of groups that the user can access and gives resources that the user can access as a result.

8 The result is sent to the Meta-PDP.

9 The Meta-PDP sends the obtained result to the PDP.

10 The PDP sends the result to the PEP.

Sr / Resources	Sr1	Sr2	Sr3	Sr4
r_1	1	0	0	0
r_2	1	1	0	0
r_3	1	1	1	0
r_4	1	1	1	1

Table 3: Security Table of Group 1

Sr / Resources	Sr1	Sr2	Sr3	Sr4
r_5	0	1	0	0
r_6	0	1	1	0
r_7	0	1	1	1
r_8	1	1	1	1

Table 4: Security Table of Group 2

Sr / Resources	Sr1	Sr2	Sr3	Sr4
r_9	1	0	0	0
r_{10}	1	1	1	0

Table 5: Security Table of Group 3

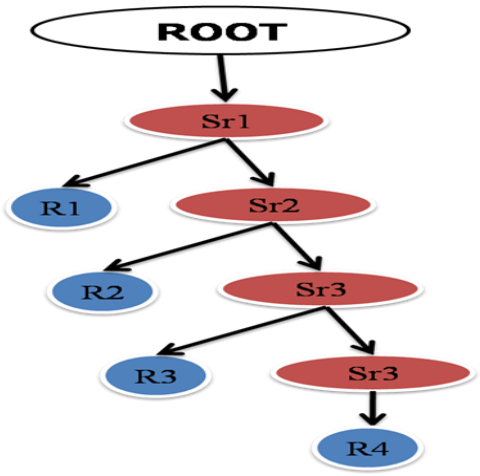

Figure 3: The Grid Authorization Graph of group 1

RESEARCH ARTICLE

Figure 3 shows the grid authorization graph of group 1.

Sr / Resources	Sr1	Sr2	Sr3	Sr4
r_{11}	1	1	1	0
r_{12}	1	1	1	1

Table 6: Security Table of Group 4

4. SIMULATIONS AND RESULTS

To demonstrate the proposed model's efficiency, a simulator was developed to manage the grid computing authorization process. This simulator is a C# application that combines both the GAG and WGAG mechanisms using the security table mechanism [11]. This table was defined as an (s*k) matrix where s represents the number of resources and k represents the number of security rules. The value of the cell (i, j) in the table can take 0 or 1 as values indicating that the j^{th} security rules belongs or not to the i^{th} resource security policy. The developed simulator used a security table of 12 resources as entries to the Grid Authorization Graph (GAG) in the first case. Four security tables were used in the second case to represent the resources security policies of each group. Furthermore, an additional security table containing the Meta-security policies of each group was created. On the other hand, a table containing all Meta-security rules degrees was created. All the previous tables were fulfilled randomly using the C# random function.

After that, the following two cases were tested:

Case one: an authorization model with a single decision point using the Grid Authorization Graph (GAG) mechanism.

Case two: an authorization model having two decision points. The first one manages the Meta-security policies using the Weighted Grid Authorization Graph (WGAG) mechanism. The second point manages the resources security policies using the Grid Authorization Graph (GAG) mechanism.

Considering a grid environment with 12 resources and 4 security rules, 100 authorization processes were initiated in the case of Grid Authorization Graph (GAG) and the proposed model. Simulation results are illustrated in figure 4 (the x axis indicates the number of requests, while the y axis indicates the complexity).

According to the results illustrated in the Figure 4, it is noticed that the complexity of the proposed model (blue line) is always less than that of the GAG (red line). This is one of the advantages of using the notion of priority to access resources. The 'priority clearance' attribute, which defines the type of resources that the user wants to access (powerful resource or not) as well as the domain to which the user belongs, allowed to reduce the size of the graph to be parsed, which explains the decrease in the number of security rules that require to be verified.

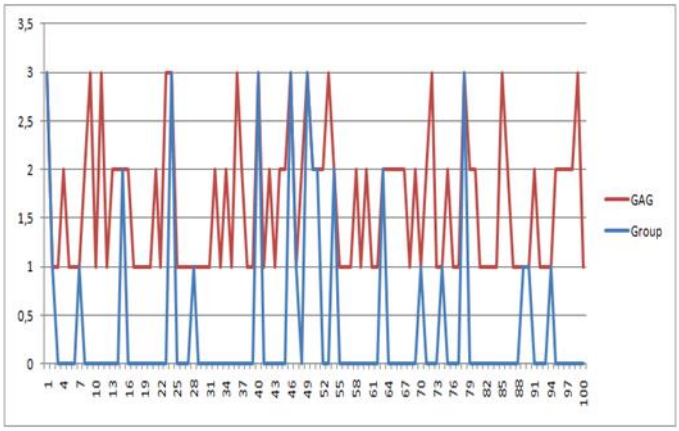

Figure 4 : Simulation Results

It is observed that the complexity of the proposed model has reached zero in some cases while that of the GAG is always greater than zero. This represents the case where the needs of the user are not satisfied by any of the groups (the security meta-policies have not been satisfied by the values of the attributes of the request).

5. DISCUSSION

In the proposed model, the resources were grouped according to their performance. The Meta-security policies principle was used because it allows for attributing the most efficient resources to the jobs requiring it. A Grid Authorization Graph (GAG) is built for each group using its resources policies. This permits to parse only the graph that the user specifies according to the resource performance adequate to its job. In addition, decreasing the size of the graph to be parsed permits to reduce the necessary time to make an access control decision. The simulation results presented in the previous section have demonstrated that the proposed model has effectively reduced the complexity. Finally, it is noticed that the proposed model guarantees a better use of resources by assigning them according to the needs of every job.

6. CONCLUSION

In This article, the two-level grid computing access control model has been proposed. This model is based on resource performance and request priority. Resources were grouped according to their performance. The Weighted Grid Authorization Graph (WGAG) and the Grid Authorization Graph (GAG) were combined to ensure an efficient access control model. Simulation results showed that the proposed model has effectively reduced the number of security rules that require verification. This enhancement permits to diminish the size of the graph that needs to be parsed which

RESEARCH ARTICLE

implies the reduction of the answer time to an access control request.

REFERENCES

[1] Anirban Chakrabarti, "Grid Computing Security", Library of Congress Control Number: 2007922355 ;ACM Computing Classification (1998): C.2, D.4.6, K.6.5,ISBN 978-3-540-44492-3 Springer Berlin Heidelberg New York (2007).

[2] Gouglidis A. and Mavridis I., (2012). Grid access control models and architectures. Computational and Data Grids: Principles, Applications and Design. DOI: 10.4018/978-1-61350-113-9.ch008.

[3] Namane, S. & Goualmi, N. (2019). Grid and Cloud Computing Security: A Comparative Survey. International Journal of Computer Networks and Applications (IJCNA) published on January 2019 (DOI: 10.22247/ijcna/2019/49572).

[4] Xiao-jun Zhu; Shi-qin Lv; Xue-li Yu and Guang-Ping Zuo, DynamicAuthorization of Grid Based on Trust Mechanism, 2010 InternationalSymposium on Intelligence Information Processing and TrustedComputing, DOI: 10.1109/IPTC.2010.113 (2010).

[5] Tiezhu Zhao and Shoubin Dong, A Trust Aware Grid Access ControlArchitecture Based on ABAC, 2010 IEEE Fifth InternationalConference on Networking, Architecture, and Storage, DOI:10.1109/NAS.2010.18, (2010).

[6] Bhavna Gupta; Harmeet Kaur; Namita and Punam Bedi, Trust BasedAccess Control for Grid Resources, 2011 International Conference onCommunication Systems and Network Technologies, DOI:10.1109/CSNT.2011.146, (2011).

[7] Kaiiali M. ; Wankar R. ; Rao C.R. & Agarwal A., (2010). A Rough Set based PCM for authorizing grid resources. 2010 10th International Conference on Intelligent Systems Design and Applications. PP. 391-396 (DOI: 10.1109/ISDA.2010.5687232).

[8] M. Kaiiali, R. Wankar, C.R. Rao, A. Agarwal, New efficient tree-building algorithms for creating HCM decision tree in a grid authorization system, in: The 2nd International Conference on Network Applications Protocols and Services, NETAPPS, Malaysia, 22–23 September 2010, pp. 1–6.(2010).

[9] Kaiiali M. ; Wankar R. ; Rao C.R. & Agarwal A., (2010). Enhancing the Hierarchical Clustering Mechanism of Storing Resources' Security Policies in a Grid Authorization System. International Conference on Distributed Computing and Internet Technology ICDCIT 2010: Distributed Computing and Internet Technology PP. 134-139 (DOI: 10.1007/978-3 -642-11659-9_13).

[10] Kaiiali M. ; Wankar R. ; Rao C.R. & Agarwal A., (2012). Concurrent HCM for Authorizing Grid Resources. International Conference on Distributed Computing and Internet Technology ICDCIT 2012: Distributed Computing and Internet Technology.PP 255-256. (DOI: 10.1007/978-3-642-28073-3_23).

[11] Mustafa Kaiiali , Rajeev Wankara, C.R. Raoa, Arun Agarwal , Rajkumar Buyyab, Grid authorization Graph , Future Generation Computer Systems 29 1909–1918 (2013).

[12] Ehsan Amiria, Hassan Keshavarzb, Naoki Ohshimab, and Shozo Komakic; Resource Allocation in Grid: A Review, International Conference on Innovation, Management and Technology Research, Malaysia, 22-23 September, 2013.

[13] Namane S., Kaiiali M., Ghoualmi N., (2017). Weighted Grid Authorization Graph (WGAG). 2017 Sixth International Conference on Communications and Networking (ComNet). DOI: 10.1109/COMNET.2017.8285589.

[14] Lawrence Chiou, Spencer Whitehead, Geoff Pilling. Graph Theory. Retrieved on October 15, 2019 from https://brilliant.org/wiki/graph-theory/.

[15] John Rushby, The Bell and La Padula Security Model. Draft Technical Note of June 20 (1986), retrieved on 15 May 2018 from https://pdfs.semanticscholar.org/ffe2/b8473a61050102f6ec7ffc6dceba98bef00f.pdf.

[16] Mustafa Kaiiali, C. R. Rao, Rajeev Wankar, Arun Agarwal, Cross-Domain, Single Resource Authorization using HCM. International Technology Management Conference, Antalya, Turkey, (2015).

[17] Grid Computing - Definition and Disadvantages. Retrieved on May 15, 2019 from http://www.brighthub.com/environment/green computing/articles/107038.aspx

[18] Yuan, E. and Tong, J., (2005). Attributed Based Access Control (ABAC) for Web Service. The 2005 IEEE International conference on web service (ICWS˝05).

[19] Bokefode Jayant. D., Ubale Swapnaja A., Apte Sulabha S., Modani Dattatray G., (2014). Analysis of DAC MAC RBAC Access Control based Models for Security. International Journal of Computer Applications (0975 – 8887) Volume 104 –No.5, October 2014.

[20] TechTarget, (2018). Mandatory access control (MAC). Retrieved on September 20, 2018 from http://searchsecurity.techtarget.com/definition/mandatory-access-control-MAC.

[21] Kaustav, R., & Avijit B., (2012). A Proposed Mechanism for Cross-Domain Authorization in Grid Computing Environment. International Journal of Emerging Technology and Advanced Engineering ISSN 2250- 2459, Volume 2, Issue 4 PP.163-166.

[22] Rissanen, E. (2013). eXtensible Access Control Markup Language (XACML) Version 3.0. OASIS Standard. Retrieved on August 20, 2018 from http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.

[23] OASIS. Open standards, Open source. Retrieved on September 20, 2019 from https://www.oasis-open.org/org.

[24] Parsing an XML File Using SAX. Retrieved on September 20, 2019 from https://docs.oracle. com/javase/tutorial/jaxp/ sax/parsing.html

[25] Wikepedia. Lightweight Directory Access Protocol. Retrieved on June 20, 2019 from https://fr.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol.

Authors

Sarra Namane obtained her Ph.D from Badji Mokhtar University, Annaba, Algeria. She is a member of Computer Networks and Systems Laboratory. Her research interests include networks security, grid computing security and cloud computing security.

Nacira Ghoualmi: is a Professor in Computer Sciences and has been a lecturer in the Department of Computer Science at Badji Mokhtar University, Annaba, Algeria since 1985. She is the Head of the Master and Doctoral option entitled Network and Computer Security, and Head of a Laboratory of Computer Networks and Systems. Her research includes cryptography, networks security, intrusion detection system, wireless networks, distributed multimedia applications, quality of service and optimization in networks, grid computing security, and cloud computing security.