AN ADAPTIVE HIERARCHICAL ACCESS CONTROL ARCHITECTURE FOR ENTERPRISE NETWORK USING COMPLIANCE VARIANCE

Sodiya A.S.

Federal University of Agriculture, Abeokuta, Ogun State, Nigeria, Department of Computer Science

Orunsolu A.A.

Moshood Abiola Polytechnic, Abeokuta, Ogun State, Nigeria, Department of Computer Science

ABSTRACT: Enterprise networks are integral components of most modern day businesses especially in today's high technology environment. The smart environment created by enterprise networks allows seamless interactions and automation. As a result, many enterprise communication netwoks require security infrastructure that ensures access permissions only to legitimate personnel. In this paper, an adaptive access control model is proposed to provide a smart method for managing, regulating, checking and ensuring only valid permissions in a hierarchical enterprise environment. The proposed adaptive access control offers a security protection framework that governs all information flow (e.g. connectivity, services, resource utilization etc.) within an enterprise network. All valid permissions and access control decisions are defined based on the security hierarchy using a privilege graph. An algorithm is developed to check compliance variance in the default access computation before access is granted. Each hierarchy of the enterprise network in the proposed approach has a membership set which defines access criteria. The access criteria are modeled using composition algebra because of its efficient policy specification characteristics. The analysis of the proposed system shows a system that can guarantee efficient control and utilization of resources in an enterprise network.

KEYWORDS: Adaptive systems, Roles, Access control, Hierarchical enterprise networks, Network Security.

1. Introduction

Modern day enterprises exhibit a growing trend toward adoption of computing services for efficient resource utilization, scalability and flexibility. This development is characterized by highly dependent heterogeneous and distributed computing systems. Consequently, this leads to exchange of enormous volume of time-critical data take with varying levels of access control in a dynamic business environment ([BGB05]). This growth comes with attendant concerns over security of enterprise resources as vulnerabilities are regularly discovered in a wide variety of enterprise software applications ([HO09]). The leak of sensitive information to fraudulent users through such security breaches can wreak havoc which can affect both customers and organizations. The implication is the loss of confidence in the usability and efficiency of the organization's computer network. In addition, the justification for investment in IT infrastructure is reduced. In the light of these challenges, the notion of confidentiality, integrity, authentication, and availability becomes a central security issue of enterprise network resource management ([SO09]). Therefore, network security is one of the first priorities of any enterprise network, where there is need for induced defense for efficient access to corporate resources ([BS13]).

There have been many attempts to make networks more manageable and more secure. To this end, enterprise network operators typically implement network security policy using middle boxes, firewalls, intrusion detection systems, and a collection of complex network configurations ([N+09]). Middle boxes can exert effective control only if placed at network choke-points. However, they become sufficiently inefficient when network traffic accidently flows around them ([CFS07]). Firewalls are generally useless against inside attacks which are a serious concern in a corporate setting ([C+03]). Authentication systems like passwords and biometrics cannot prevent legitimate users from carrying out harmful operations on a corporate application. Instead of deploying trust on the servers and relying on security middle boxes, an enterprise network should offer a filtering system that controls network traffic to different levels of an enterprise network ([N+09]). This is because it is important to control the flood of information within an enterprise network in an attempt to capture security threats and vulnerabilities.

Access control is the process of mediating every request to resources and data maintained by a system and determining whether the request should be granted or denied ([***95]). Access control plays a central role in efficient management of enterprise network because of its support for robust policies representation, permission specification, enforcement monitoring etc. The enterprise networks need systems that determine who gets access to specific information ([AD03]). These systems must be founded on a chain

of trust stipulated by explicit rules and roles. An efficient access control system records and timestamps all communications and transactions so that access to enterprise network and information system can be audited later. In an enterprise application, group members subscribe to different resource streams, or possibly multiples of them. Thus, it is necessary to develop group access control mechanism that supports the multi-level access privileges, referred to as the hierarchical access control ([SR04]).

The aim of this paper is to develop an Adaptive Hierarchical Access Control Architecture (AHACA) using compliance variance computation for an enterprise environment. The rest of this paper is organized as follows: Section 2 presents literature review. The architecture and methodology of AHACA is described in Section 3. In Section 4, an analysis of the model using case scenario is presented and Section 5 presents the concluding remarks and future works.

2. Literature Review

Access control system is one of the most recurrent topics in information security. The need to protect enterprise information system from internal and external threats has widened the scope of access control systems, models, and implementation ([SPK09]).

Mudtadi et al. ([MHA10]), investigated access control in ubiquitous computing environments using threshold cryptography and multilayer encryption to provide dynamic and truly distributed method of security control. The architecture of the approach uses policy service, context service, and event service. Ardagna et al. ([A+11]) used policy spaces for access control in healthcare environment where the authors finely depicted the exceptional case of the break-the-glass scenarios. The authors used algebraic representations for the different policy spaces and access regulations to data. Sandhu and Zhang ([SZ05]), studied trusted computing technologies in the domain of access control practices for peer-to-peer environment. However, the solution proposed by the authors requires a fully protected runtime environment to ensure the trustworthiness of the application. This makes the practical implementation of this approach costly as pointed out by Han et al. ([H+10]).

Ghadi et al. ([G+09a, G+09b]), investigated hierarchical role graph model for the UNIX access control system. The model is based on super-user model and role- based access control model in which the notion of privilege graph was used to build the hierarchical system. The properties of graph theory were used to evaluate the stability and the robustness of the model.

Yan and Ray ([SR04]) studied the security infrastructure in a multi-level group communication. The authors presented a multi-group key management scheme that achieved hierarchical access control by employing an integrated key graph and by managing group keys for all users with various access privileges. The approach significantly reduces the communication, computation, and storage overhead associated with key management and hierarchical systems.

Homer and Ou ([HO09]) presented an approach based on Boolean Satisfiability Solving that can reason about attacks, usability requirements, cost of actions, and other relevant parameters in an enterprise network security. The approach presented a balance between security and usability of an enterprise network. Nasirifard et al. ([NPD11]) provided a comprehensive framework for annotation-based access control for collaborative information spaces such as social networks. The authors presented a collaboration vocabulary to finely express the annotation approach. Although, the work is demonstrated for social networks whereby access to resources is tagged, the authors did not consider cases where annotation creates abuses forcing the resource owner to withdraw access.

Kabir et al. ([KWB11]), proposed conditional-purpose-based access control model with dynamic roles. The approach provides a dynamic support for the traditional role-based access control to ensure privacy of data and information.

3. Overview of AHACA Architerture

The architecture of AHACA is a spatial arrangement of various components used in actualizing the proposed access control model (Figure 1). It is a two-region based architecture consisting of entities and corporate resources. The entities region registers and specifies user access to corporate resources which is determined by a default calculated value called Default Intelligence Trust Value (DITV). DITV is defined as a non-empty set whose values are preset to true or false depending on the permission level indicated in the policy specification. The corporate resource region authenticates, authorizes and determines user access to resources when all conditions in policy specification evaluates to positive. The access verification is a thin layer that separates the two regions in the AHACA architecture. The access verification checks for access compliance before an entity is transited into the corporate resource region.

3.1 Stages in AHACA Methodology

The proposed AHACA methodology is a multi-stage approach which (i) registers and specifies types of user access to a shared corporate resources (ii) authenticates and authorized access to a pre-registers

resources (iii) determines compliance of registered users to resource usage based on access policy.

Stage 1: Access Registration and Specification

Access Registration and Specification is the first phase in the proposed hierarchical access control architecture explained as follows:

Let S = $\{h_1, h_2, \ldots h_n\}$ be security hierarchies defined on an enterprise network *E*. For each level $h_{i=1,2\ldots,n \in \mathbb{Z}^+} \in S$, a relation $\mathcal{R}$ is defined as follows: Suppose that $h_i \mathcal{R} h_{i+1} \Rightarrow h_i > h_{i+1}$, then the set S is a partially ordered set where security hierarchies are reflective, anti-symmetric and transitive. Thus, security hierarchy h_1 is the highest security hierarchy level while h_n is the lowest security hierarchy level.

Definition 1: A membership set *M* is a set of subjects, *s*, (for instance, employees).

$M = \{s_1 1\ s_1 2\ s_1 3 \ldots s_1 j\}_{j=1,2,3\ldots \in \mathbb{Z}^+}$ where set *M* is a domain which forms a many-to-one mapping with elements of *S*.

Definition 2: A privilege graph *PG* is a 4-tuple of the form:

$PG = \{object, rules, role, permission\}$

which describes how roles represent a set of privileges on an object. The domain *PG* forms a one-to-one mapping with different images (.i.e. security hierarchies) in *S*. A subject, *subj*, is an entity within a particular security hierarchy h_n that can perform an action on an object. An object, *obj*, is any entity (e.g. enterprise resources) on which certain actions are performed (e.g. write, delete, update etc) depending on the rules and roles of the subject. Rule is a procedure that defines how authority is administered on an object. Role is a title which defines an authority level of any $s \in h_n$. Permission is the ability to perform some action on some object.

Definition 3: Let $M \times S = L$ *and* $PG \times S = K$. Then, the set $G = L \times K$ is an access policy specification space $\forall s \in E$.

Definition 3.1: Role/Rule assignment: A subject can execute a transaction if the subject has selected or has been assigned a role according to a specified rule.

Definition 3.2: Role/Rule authentication: A subject's authentication to a particular corporate resource is a measure of his active role and rule. Authentication, *Au*, is modeled as:

Au= f (*role* $\cup$ *rule*)

Definition 3.3: Transaction authorization *TrAu*: A subject can execute a transaction if the transaction is authorized for the subject's active role. The user login constraint, *us*, connection establishment constraint, *ce*, information control constraint, *ic*, and user role constraint must evaluate to true before a subject can access a resource. The presence of these conditions also has great implication on the security of transaction, which is the main concern of any access control systems. Thus,

TrAu =

$(us \rightarrow 1, ce \rightarrow 1, \quad ic \rightarrow 1) \cap (f(role \cap rule \rightarrow 1))$ …(i)

For each *subj*, *s* the algorithm 1 applies:

Start: $\forall\ subj, s \in E$

{ *if new then*
Register
else Check credentials
{
if (check s.credentials = s.setcredentials)
{*assign s to* h_i
extract the privileges
create rules and roles
s.Au=$\cup_{r-re}$
s.TrAu=$\cap_{r-re}$ }

elseif (s.credential ≠ s.setcredentials)
{
logout if n > 3
where n = number of attempts
}}

End

Algorithm 1: Transaction Authorization Algorithm

These sets are created by the enterprise network administrator with roles, rules and privileges clearly indicated for efficient resource authentication and authorization.

Stage 2: Access Authentication and Authorization

Access Authentication and Authorization is the second phase of the proposed architecture. The enterprise network administrator creates a controller for each h_i to manage access authentication parameter and policy *P*. The policy is a composition of a set of authorizations modeled using composition algebra. Three operators are introduced to characterize our algebra ([B+02]). For example, addition (+): this operator models the disjunction of two policies. Given A1 and A2 as depicted in the architecture, P = A1+A2 means that an access is granted if at least one of the conditions is satisfied. Conjunction (&): the operator models the intersection of two policies before access is granted. For example, P= A1&A2 means that access is granted if the two conditions are satisfied. Subtraction (-): this operator models exception where a policy restricts another policy. For example, P=A1-A2 means access is granted if A1 satisfied and A2 is not satisfied.

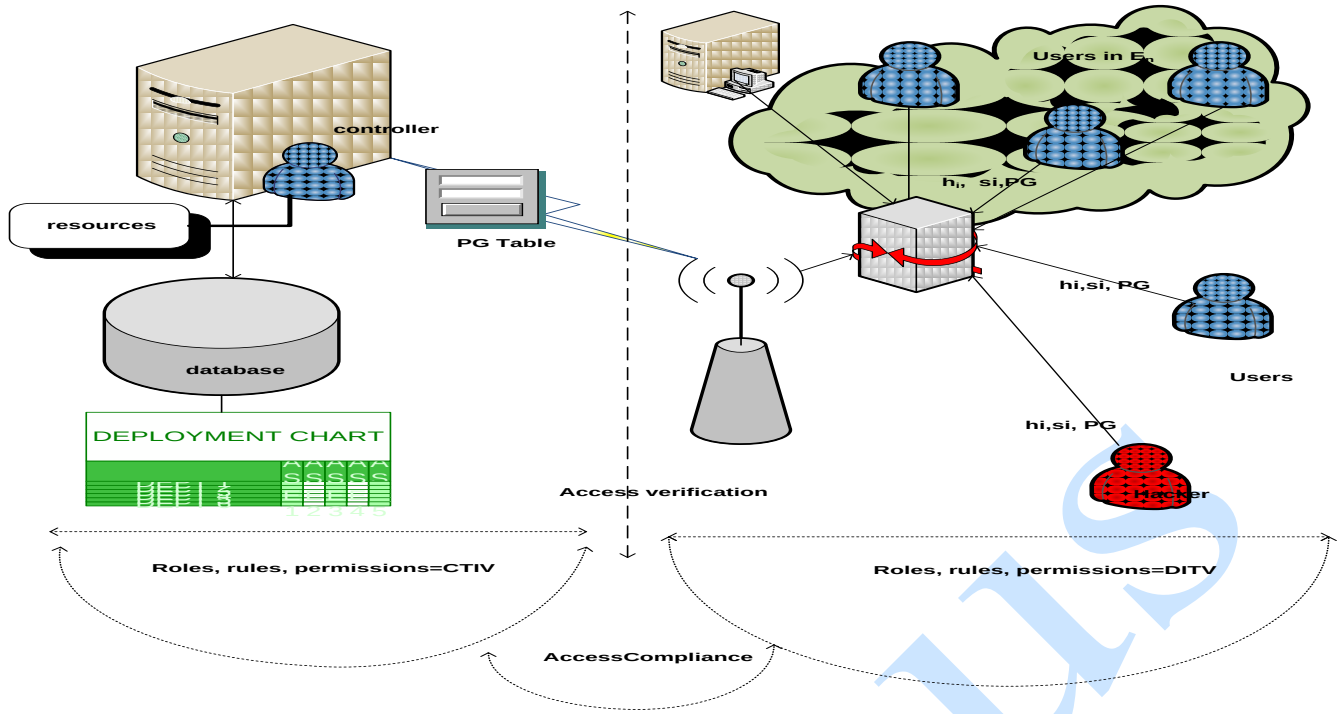

Figure 1. System Architecture

The configuration of authentication services by controller is explained in the steps below:

- The local controller checks if obj_i is within LAN
- **If** ($s.LAN == true$) ***then***
  - The local controller checks A1=$s._{r\text{-}re}$ and A2= password
  - **If** $(A1 \& A2) == .true.$ **then** AccessGranted () ==.true.
    Else *Accessdenied ()*
  - Update sAu *and* $s.TrAu$ file
- **ElseIF** obj_i in another LAN
  - The *s.LAN* local controller passes request to global controller, which oversees other controllers.
  - The global controller uses global ACL(access control list)
  - Determine if **s.LAN** controller can access obj in the other LAN
    - **If** *s.LAN*== true
      - **Then** *obj.LAN* controller grant access for *s*

Else
AccessDenied ()

Stage 3: Action Determination and Access Compliance

Action Determination and Access Compliance is the final phase of the proposed hierarchical access control system. Upon successful authentication of *s*, a Monitor Manager (MM) is invoked to check user compliance with privileges and permissions defined on the resources being accessed.

Definition 3.4: The purpose of intelligent computation, Default Intelligent Trust Value (DITV) is defined as a non-empty set whose values are preset to true or false depending on the permission level indicated in the policy specification.

Definition 3.5: Calculated Intelligent Trust Value (CITV) is defined as a non-empty set whose members indicate the action of a particular subject on a resource or set of resources on which permission has been defined.

Definition 3.6: AccessCompliance() is a function which is invoked when DITV-CITV==0 and AccessNonCompliance () is a function which indicates the existence of significant difference between DITV and CITV.

The DITV is evaluated against the CITV to determine the violation of access on a particular enterprise resource. Entity permission is computed for all the actions associated with his level and resource requested, if the DITV and the CITV returns logic value false, then access is denied. The difference between the DITV and CTIV is defined as Compliance Variance CV, which determine the severity of violation. However, AccessCompliance () is still invoked if CV is within the threshold value, which measures the degree of fault tolerance level of the system. The existence of DITV is the main contribution of this work in checking grave violation of access requests and the algorithm 2 for the scheme is demonstrated:

Start:
Input: subject, roles, permission, distanceId (optional, only necessary from outside *E*)
Output: *AccessCompliance*, *AccessNonCompliance*
Begin: $\forall\ subj, s \in E$ **do**
Retrieve list of resources from database
Check authentication parameter and policy specification
If ($.Au\ .AND.\ s.P == false$) **Then**
$Exit(\square)$
Else continue
Check (resource. policy for resource $e \in h_{is}$
End do
Response block:
DITV

$$= \sum_{I}^{N} Action\ (read + write + \cdots + update)$$

If
$(P1 = CV).AND.(DITV - CITV) == false$
If (P2 = CV≠0) **then**

$$(x + a)^n - \sum_{k=1}^{n} \mathbf{Action}$$

Threshold check () = $\leq \frac{1}{8} ft$
where *n* = no of attempt, *x* = permission, *a* = error value and ft is the fault tolerant value defined on the system by the administrator.
IF (P1&P2==. ***TRUE***.) **THEN**
AccessCompliance()
Else
AccessNonCompliance()
End if
End if
End

Algorithm 2: Access Compliance Algorithm

The controller examines the log submitted by the monitor manager. Every secure transaction should satisfy the policy constraint $\dot{s} = s$ where $\dot{s}$ policy specification and *s* is the user transaction. If violation occurs, the controller suspends obj_i access and performs the following tasks:

- The controller request for veracity of violation
- If the response generates a logical true upon computation of some other privileges of obj_i, roles and rules are updated for obj_i.
- Otherwise the obj_i r-r is blacklisted

A privilege graph manager, *gm*, is used to manage communications among the h_{is} in the model. It is the principal controller for each h_i. The access right of other controller is reserved by *gm*. A review controller provides the overall access control configuration. The review controller examines access privileges based on the log submitted by other *gm*. This controller belongs to the highest *hi* in the enterprise network. It is the chief access control manager responsible for review, violation, commission, and omission of business resources. The review controller adds, deletes, updates, and stores privileges of other controller based on the log submitted for proper auditing and vulnerabilities.

4. Analysis of the Model using Case Scenario

Consider a simple scenario of HpNoki Enterprise Network (HEN) in order to clarify the basic concepts in our proposed access control model. The HEN consists of five operational hierarchical levels namely BoardOfDirector, TopManager, Operations, Accounting and Personnel. Every employee in HEN is assigned to a specific security hierarchy *h* defined on HEN this security hierarchy *h* indicates the employee access states, transaction, constraint etc. as contained in policy specification of HEN. Initially, considers an employee with identity number, *emp.Id*, who belongs to TopManager level, initiates a read operation in accounting hierarchy when preparing budget and operational expenses to the BoadOfDirector. The employee notices some irregularities and calls E2 exception from operational environment policy space, *op_env*, to initiate a query action. As a consequence, the employee can inform their auditing firm to make connection to HEN for investigation and verification of accounting transactions over a period of time. However, since this connection is new or has infrequent connection (since auditing is a routine business exercise) with HEN, the principal controller checks the membership set of this connection and it will definitely evaluate to false. Instead, exception *E3* in the *op_env*, is applicable and evaluate to true (Figure 3). The controller creates the registration phase (user login constraints), authentication phase (connection establishment constraints), scanning phase (information control constraints), and operation phase (user role constraints) for this new connection. The registration phase installs the membership set and access privileges for this connection, the authentication phase verifies this connection during the period of operation with the principal controller, the scanning phase cleans the connection of potential vulnerabilities which can steal sensitive business information, and the operation phase allows access for intended task. Note that the equation (i) must be satisfied before access can proceed. If during the operation phase, the compliance computation variance returns false against default intended task in the registration phase, the access granted is suspended and the auditing firm network connection is blacklisted. In case of internal operation, access is

evaluated against condition in E1 for normal operation, E2 for abnormal operation, and E4 for emergency. These operation environment constraints have advantages of ease of policy updates, revocation, administration, layering consideration, efficiency of policy evaluation, simplicity, and security.

Table 1. An example of operational environment policy space specifications

	op_env	Rule	Description
E 1	State:=normal operation	User.role = emp ^ empId = userId, emp.DutyHours = time(), object.type = business info{read, write,….}	An employee can read and access business information under his responsibility specified by his membership set
E 2	State:=abnormal operation	User.role=emp.hi > emp.h0, object.type = transaction data Object.action = {read}^query	A senior employee can read transaction data of lower department and initiate query in case of irregularities
E 3	State:=auditing operation	User role = s.id ^ object.type = account data ^ object.action = {read}, s.id # empId	An auditor can read the account data of any department in case of auditing exercise
E 4	State:=emergency	User.role = emp ^ empId # userId, emp.DutyHours = time(), object.type = business info{read}	An employee on duty can read any company data not under her responsibility in case of emergency

5. Conclusion

The hierarchical approach captures the natural flow of authority within an enterprise network. In an enterprise network, the management of resources is becoming an increasingly challenging problem in no small part due to scaling up of measures such as protocols, applications, network elements, number of users and functionality expectations. To manage access control in these networks and deliver the required services, intelligent tools and architectures are needed to cope with the complexity of the network entities and their respective policies for interaction. In this paper, the concept of intelligent trust computation is introduced into a hierarchical enterprise environment before access is fully granted to network resources. In addition, analysis of compliance computation variance algorithms is provided. The use of algebraic primitives for policy specification in the proposed approach allows for consideration flexibility in policy combination. This technique presents a more helpful method that make internal control, resource auditing and enforce privacy of shared corporate resources.

References

[A+11] **Ardagna C., Capitani S., Vimercati D., Foresti S., Grandison T., Jajodia S., Samarati P.** - *Access control for smarter healthcare using policy spaces*. Journal of computers and security, 848-456, Elsevier Ltd., 2011.

[AD03] **Austin R. D., Darby C. A. R.** - *The Myth of secure computing*. Harvard Business Review, 120-126, 2003.

[BS13] **Barbole K. N., Satav S. D.** - *Next Generation Firewall in Modern Network Security*. International Journal Data and Network Security. Vol 3. No 2, 2013.

[BGB05] **Bhatti R., Ghaffoor A., Bertino E.** – *X-GTRBAC: An XML-Based Policy Specification Framework and Architecture for Enterprise-Wide Access Control.* ACM Transactions on Information and System Security, Vol. 8, No.2, 2005.

[B+02] **Bonatti P., De Capitani S., di Vimercati D., Samarati P.** - *Algebra for composing access control policies*. ACM Transactions on Information and System Security 5, pp 1-35, 2002.

[C+03] **Cai Z., Guan X., Shao P., Peng Q., Sun G.** - *A rough set theory method for anomaly intrusion detection in Computer network system.* Journal of Expert System, 2003.

[CFS07] **Casado M., Freedman J., Shenker S.** - *Ethane: Taking control of the Enterprise*. SIGCOMM '07. ACM, 2007.

[G+09a] **Ghadi A., Mammass D., Mignotte M., Sartout A.** - *Formalism of the access control model based on the Marked Petri Nets*. International of Journal of u- and e- Service, 2009.

[G+09b] **Ghadi A., Mammass D., Mignotte M., Sartout A.** - *Hierarchical Role Graph Model for UNIX Access Control*. International Journal of Control and Automation, Vol.2., 2009.

[HSR08] **Hagen J.M., Siverteen T.K., Rong C.** - *Protection against unauthorized access and computer crime in Norwegian enterprises*. Journal of Computer Security 16, 341-366, IOS Press, 2008.

[H+10] **Han W., Xu M., Zhao W., Li G.** - *A trusted decentralized access control framework for the client/server architecture*. Journal of Network and Computer Applications 33, 76-83, Elsevier, 2010.

[HO09] **Homer J., Ou X.** - *SAT-Solving Approaches to Context-Aware Enterprise Network Security Management*. NSF, 2009.

[KWB11] **Kabir M. D., Wang H., Bertino E.** - *A conditional purpose-based access control model with dynamic roles*. Journal of Experts Systems with Applications, 1482-89, Elsevier Ltd, 2011.

[MHA10] **Muhtadi J., Hill R., Al-Rwais S.** - *Access control using threshold cryptography for ubiquitous computing environment*. Journal of King University-Computer and information Sciences,71-78, 2010.

[NPD11] **Nasirifard P., Peristeras V., Decker S.** - *Annotation-based access control for collaborative information spaces*. Journal of Computers in Human Behavior, 1352-64, 2011.

[N+09] **Nayak A., Reimers A., Feamster N., Clark R.** - *Resonance: Dynamic Access Control for Enterprise Networks*. WREN, Barcelona, Spain, 2009.

[Ric76] **Richard D.** - *Hierarchical organization: a candidate principle for ethnology*. Cambridge, England, 1976.

[SZ05] **Sandhu R., Zhang X.** - *Peer-to-peer access control architecture using trusted computing technology*. In Proceedings of the 10th ACM symposium on access control models and technologies (SACMAT '05), Stockholm, Sweden, ACM Press, p. 147–158, 2005.

[S+96] **Sandhu R. S., Coyne E. J., Feinstein H. L., Youman C. E.** - *Role-based access control models*. IEEE Computer, 29(2), 38–47, 1996.

[SO09] **Sodiya A.S., Onashoga A. S.** - *Components based access control Architecture*. Issues in informing science and information technology. Vol. 6, 2009.

[SR04] **Sun Y., Ray Liu K.** - *Scalable hierarchical Access Control in Secure Group Communications*. IEEE, 2004.

[SPK09] **Singh M., Patterh M., Kim T.** - *A Formal Policy Oriented Access Control Model for Secure Enterprise Network Environment*. International Journal of Security and its Applications, 2009.

[ZEW08] **Zhang B., Eugene T., Wang G.** - *Reachability Monitoring and Verification in Enterprise Networks*. SIGCOMM, 2008.

[***95] **NIST** - *An introduction to Role-based Access Control*. Bulletin of NIST, 1995.