International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 7, No. 3, September 2022 – November 2022 | Pages 46-54

ENHANCING CLOUD SECURITY: INTEGRATING ATTRIBUTE-BASED ACCESS CONTROL WITH OPENSTACK'S EXISTING RBAC FRAMEWORK

Dr. A N Nandakumar, Feby Ashraf, Dr.B.Sivakumar,Nice Mathew,
Department of Computer Science and Engineering (CSE)
Indira Gandhi Institute of Engineering and Technology
Nellikuzhi P.O, Kothamangalam, Ernakulam (Dist), Kerala, Pincode 68669, India

Abstract:
Cloud computing is currently regarded as a leading paradigm within the Information Technology sector. It delivers innovative cost-effective services on demand, including Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS). Despite the substantial advantages these services offer, cloud computing faces several challenges such as data security, service abuse, malicious insiders, and cyber-attacks. Access control remains a critical security requirement in cloud computing to prevent unauthorized access and protect organizational assets. While various access control models and policies have been developed for different environments, they often do not meet the specific needs of cloud access control. This study integrates components of the PM framework with a proof-of-concept implementation to deploy an ABAC extension for OpenStack, maintaining the existing RBAC architecture. This approach enhances access control flexibility through the support of user attributes while minimizing the disruption to the existing OpenStack access control framework. Use cases are provided to illustrate the additional benefits of the proposed model and demonstrate the authorization results. The performance of the ABAC extension is evaluated, discussing its practicality and potential performance improvements.
Keywords: Attribute-Based Access Control, Model, IaaS, Cloud

1. Introduction

Attribute Cloud-based IaaS access control

Barriers are crucial and indispensable tools for promoting enhanced levels of approach safety. As an example, an organization's objectives may set down overarching regulations to restrict the authority of delegated individuals, such as limiting the roles of "computer programmer" and "analyzer" to the same project.

Ultimately, this foundation hinders the worker's ability to simultaneously work on and test code for the same project. This concept suggests incorporating specific need information into attribute-based access control (ABAC) and cloud infrastructure as a service (IaaS), particularly in relation to ABAC. The primary objective is to closely monitor customer consent or the subject's access to the framework's resources. According to the criteria for approval that are linked to a certain authority.

Virtualization technology is a crucial component in the provision of Infrastructure as a Service (IaaS). Consequently, the hypervisor has an impact on the effectiveness of the access control architecture.

The hypervisor poses a security vulnerability with the management of access to virtual machines (VMs) owing to its centralised access point. An untrusted hypervisor poses a greater

chance of failure for a trustworthy virtual machine compared to a trusted hypervisor hosting an untrusted virtual machine.
For instance, after doing observational research on the migration process of a real-time virtual machine (LVMM), it is advised to assess the existing access control architecture in order to avoid unauthorised access during the LVMM process.
The interaction between the Logical Volume Manager (LVMM) and the virtual machine is seen as a crucial cycle in Infrastructure as a Service (IaaS) and takes place right after the virtual machine provisioning process.
In Infrastructure-as-a-Service (IaaS), access control security solutions like firewalls and security clusters lack the capability to facilitate the implementation of security-aware policies or procedures. Due to the risk of unauthorised access to sensitive data, significant efforts are made to enhance the balance between data flow security and the flexibility of Infrastructure as a Service (IaaS). A well-designed IaaS access control solution is essential in this setting.

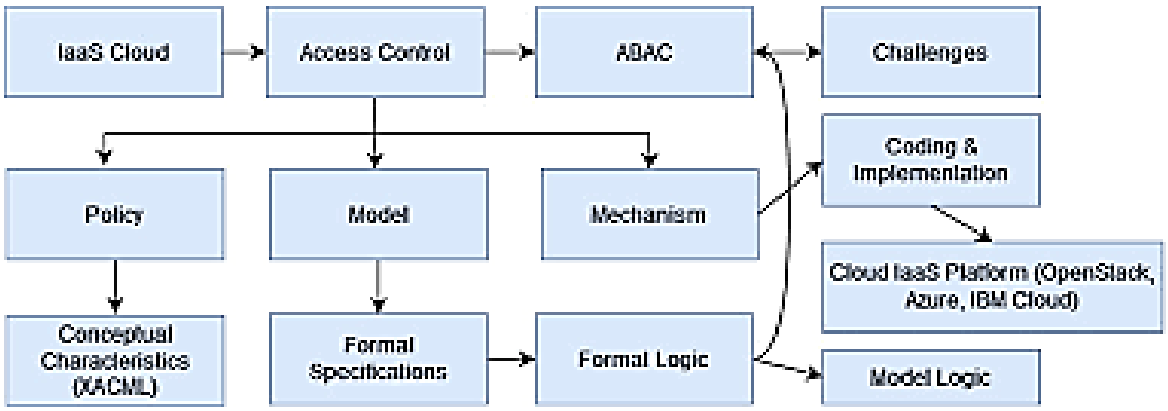

Fig. 1. General components of the investigation

The research emphasises that security issues play a crucial role in persuading clients to use cloud computing services. According to a poll conducted by IDC, 87% of customers said that their primary reason for utilising cloud computing management is their satisfaction with the degree of security and safety. The security of the IaaS cloud system is primarily concerned with controlling access to resources. The IaaS cloud differs from conventional computing environments by offering explicit capabilities such as flexibility, multi-tenancy, configurability, and dynamics, among others. Traditional access control approaches encounter difficulties in terms of flexibility and granularity while being implemented and configured in IaaS.

2. Related Work

The paper discussed the problems related to controlling access in Infrastructure as a Service. Implementing the access control measures shown in Figure 2 is considered to be the preferred method for addressing some security concerns related to Infrastructure as a Service (IaaS). Due to the fact that IaaS operates in a multi-tenant environment, it is required to fulfil a diverse set of client access requirements. Hence, it is essential to build the IaaS access control system in a manner that enables precise and detailed policy execution.
Nevertheless, ABAC does not exhibit pop-up highlighting in this particular situation. In same conditions, the designer developed ABAC to provide setup recommendations specifically tailored to integrate sophisticated RBAC models. Smari et al. expanded the scope of ABAC to include the relationship between the environment and access control, as well as concerns and

objections. This includes identifying framework conditions, protection, and trust. Both the assessment of the intricacy of the framework and the formulation of the strategic language have not been thoroughly delineated so far. The HGABAC model, suggested by Servos and Osborn, replaces the abacus with hierarchical components in the ABAC paradigm. Servos and Osborn included environmental considerations into their design by examining the relationship between organisation and environmental elements.

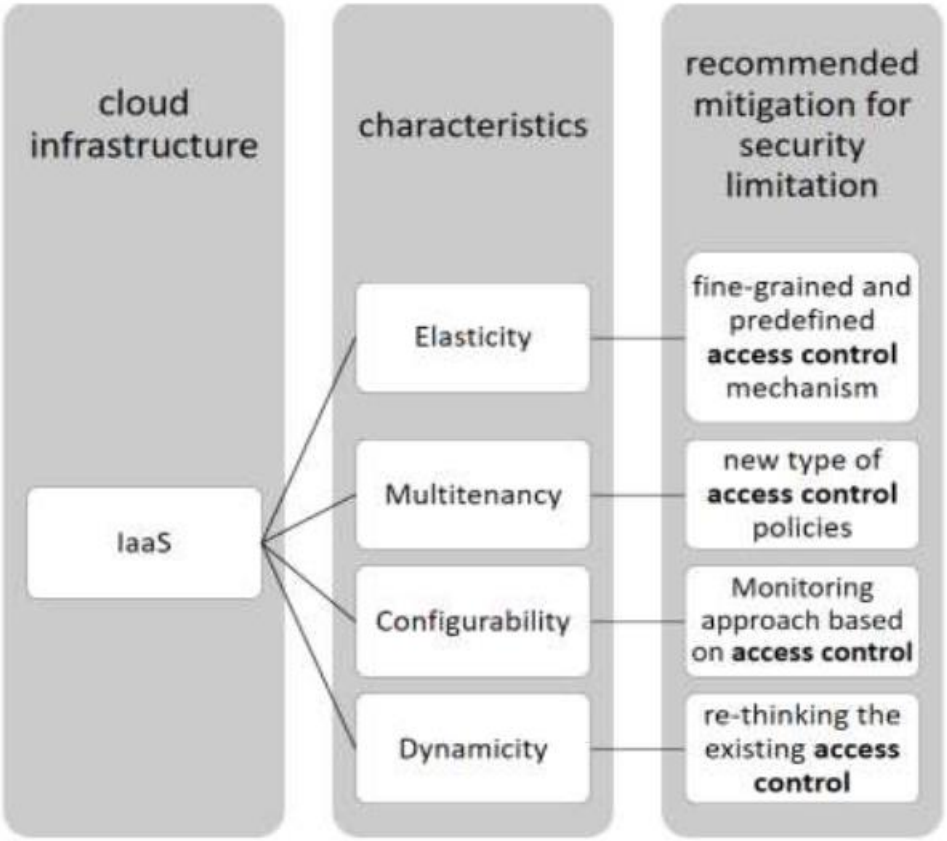

Figure 2 illustrates the suggested measures to address particular security problems in the Infrastructure as a Service (IaaS) environment.

2.1 ABAC in the IaaS cloud

The ABAC paradigm incorporates the idea of attributes, allowing users to choose rights based on these qualities.
Nevertheless, ABAC fails to exhibit pop-up highlighting in this particular situation. In same conditions, the designer devised ABAC, which provides configuration recommendations specifically tailored to integrate sophisticated RBAC models. Smari et al. expanded the scope of ABAC to include the relationship between the environment and access control, as well as addressing concerns and objections. This includes emphasising framework conditions, protection, and trust. Thus far, there has been no detailed blueprint provided for either the assessment of the intricacy of the framework or the formalisation of the strategic language. The HGABAC model, suggested by Servos and Osborn, replaces the abacus with hierarchical components in the ABAC paradigm.

Servos and Osborn included environmental considerations into their design by examining the relationship between organisation and environmental characteristics. However, they failed to consider the notion of destination and the focus on delegation.

Formal logic is used in Attribute-Based Access Control (ABAC) with a focus on version 2.2. The primary language used in HGABAC was designed taking into account the principles of Kleene K3. The languages used in ABAC and HGABAC are a kind of propositional logic. During a strategy update or strategy research, you encounter the challenge of NP-complete satisfaction. Regardless of the specific language employed, any conventional language that is based on the logic of the first criterion would encounter an undecidable computational job when attempting to explain the ABAC approach. Wang et al. created a programming language specifically designed for analysing requirements. They then used this language to construct a meaning-based framework for Attribute-Based Access Control (ABAC). This framework is notable for being the first commercially applied implementation of its kind. Bijon and his colleagues, in their article titled "ABCL," established a fundamental language specifically designed to describe the intricacies of attribute-based restrictions. Therefore, the essential language is clear and direct, since it emphasises the specific desired advantages of the capabilities found in the ABAC model, rather than using indirect or implied terminology. However, ABCL was unable to evaluate the advantages of the functioning of the authorization policy rules.

2. 3 An ABAC Extension for Openstack

The proposal suggests an extension of the established OSAC architecture to create a role-based ABAC model for OpenStack.

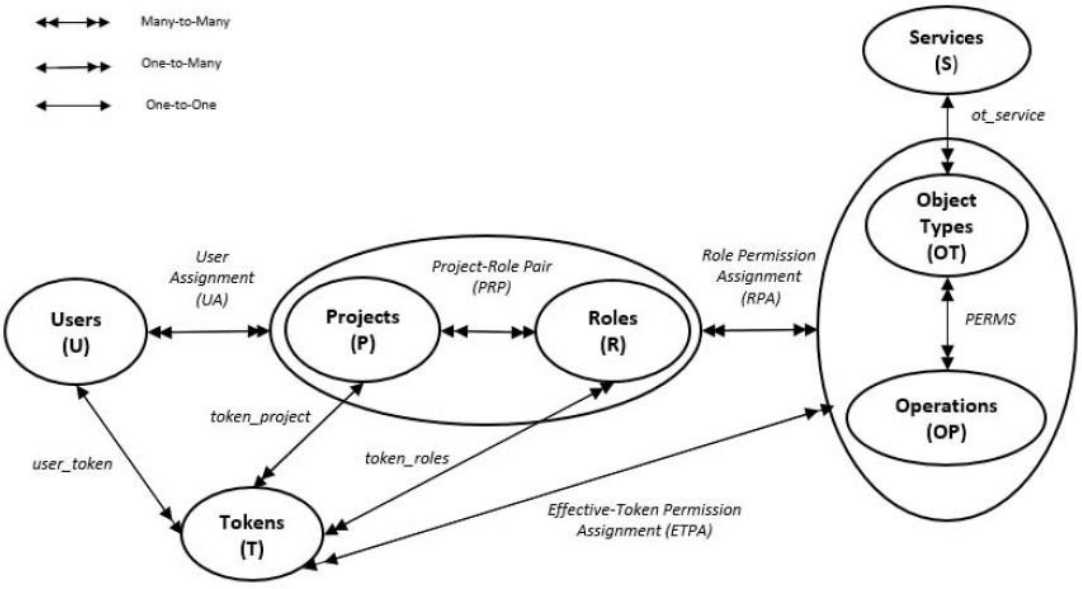

Figure 3: OSAC with extended user attribute on a single tenant

The model shown in Figure 3 is referred to as the Extended OSAC Model with User Attributes. The expanded OSAC model includes all essential and stated components, along with newly included aspects and linkages. The user table is enhanced with client attributes, and a new UAPA connection is established to get the client attribute values and carry out authorization operations. The ABAC role-based paradigm is referred to as such because it seamlessly incorporates the existing OpenStack RBAC system while preserving all of its advantages.

Additionally, it provides the versatility of an Attribute-Based Access Control (ABAC) paradigm by presenting customer attributes. The client's duties dictate the most stringent permissions, which are then further restricted by the client's attribute authorization actions.

Attribute functions are functions that accept the client as input and produce a result that is exclusive to a certain scope. The scope of an attribute refers to a limited collection of individual characteristics that are unique to each attribute function. Generally, there are two categories of anticipated characteristics: estimated atomic attributes provide just a single value within their range, whereas defined predicted attributes provide a subset of the values within their range. client attributes refer to the distinct qualities or traits of a client. Examples of these attributes are Department, Authorization, and Specialisation, which serve as representative models.

OSAC's expanded user attribute paradigm restricts clients to only have attributes that are assessed atomically. UAPA refers to a collection of authorizations linked to consumer traits and features that are issued to them. The ETPA was modified to include authorizations for customer properties, irrespective of whether the client has granted consent.

2.4 Cloud Services

Software as a Service (SaaS) is a prominent advantage of the forthcoming cloud computing technology. Infrastructure as a Service (IaaS) PaaS (platform as a service) Infrastructure as a Service (IaaS) Each of these cloud services requires consumers to acquire a kind of administration, such as programming or cycle planning. This may be either free or paid on a per-use basis if you are using a pay-as-you-go approach.

Software as a Service (SaaS)

Software as a Service (SaaS) is a cloud-based model where software applications are delivered and used over the Internet. The end user may access it using software that serves as a point of engagement. This configuration enables a method in which there is no need to install any software on the machines of clients. Customers are relieved of the need to buy, upgrade, or manage the software they use, while support providers handle these tasks. Examples of Software as a Service (SaaS) include Google Apps and negotiation software.

Platform as a Service (PaaS)

Stage as a Service offers clients the essential infrastructure to create software applications on the Internet.

These tools may be used in cloud computing and can be accessed via a web-based application. Professionals that use PaaS to construct their apps may have several advantages, such as a decrease in infrastructure expenses that would typically arise during the creation of the programme. Furthermore, it consolidates several administrations such as business intelligence, databases, middleware, and more into a unified platform. Learn is an instance of Platform as a Service (PaaS).

Infrastructure as a Service (IaaS)

End users are offered archiving, mounting, and administration services as components of the infrastructure-as-a-service paradigm. They may be obtained upon request. It offers the necessary framework for users to transmit and execute their programmes across a network. This decreases the force pushing upwards to uphold infrastructure for the advantage of consumers. For instance, Amazon Web Services and Google Compute Engine are two specific instances of Infrastructure as a Service (IaaS). This text outlines the many services offered by different cloud providers, while also discussing the responsibilities of maintaining these

resources under different support models, regardless of whether it is the service provider or the client.

Goals

1. Research the characteristics of cloud-based Infrastructure as a Service (IaaS) access control. The topic of investigation is the study of access control mechanisms in the context of cloud computing.

3. Methodology of Research

Cloud Computing Access Control (AC3)

The suggested model aligns with the job and racial requirements that have previously been put out in the earlier study. Customers are categorised in the model based on their current circumstances.

By doing so, customers are put in a secure environment that aligns with their specific job description and duties. Every position in the model is allocated a collection of the most pertinent and essential activities to do in order to be ready for that specific function. Every impacted action is allocated a security group for data or resource access, along with the precise permissions necessary to carry out the activity. A betting machine allows for the manipulation of the customer's random, dynamic, and unexpected behaviour. It also enables the solicitation of credits from customers depending on their chosen means of accessing the game.

Furthermore, a security label engine is used to implement security labels in environments that are partially trusted or not trusted, and throughout several cycles. Access to data or resources may be limited by designating security names for certain data or resources in the model. When accessing the data, it is important to prioritise the attributes of the action above the security names of the data or resources. Our proposal involves the use of security labels in certain scenarios, which are determined by the amount of trust and security that exists within the ecosystem. The suggested security token, seen in Figure 4, would consist of the client's role, command, rights, current area, defined time, and an irregular exception number that would possess irregular characteristics.

AC3 comprises the fundamental components listed below:

- Customers (U) refers to a collective of individuals who are patrons of a business or organisation.
  Roles(R)e is a compilation of many roles.
- Tasks (T) refers to a collection of specific activities or assignments. Sessions (S) is a compilation of separate sessions.
- Permissions (P) is a compilation of several permissions. The data (D) is a compilation of information.
- The Customer Assignment (UA) e is a subset of the convergence between U and R. It is a subset of the convergence between U and R.
- Role assignment (RA) is a subset of convergence between R and T. The set PAe is a subset of the convergence between sets P and T.
- The permission assignment (PA) e is a subset of the convergence between P and T. The collection of limitations in the framework, such as the division of tasks and the delegation of powers, is denoted by the symbol e.

- In the model, categories (Cla) refer to a collection of security classifications that are used to systematically arrange operations.
- Confidentiality Tokens (SL) refer to a collection of confidential names that are used to limit data access according to the data's level of sensitivity. A group of security labels, referred to as ST, is known as a security label collection.
- Nevertheless, there is a single exception from this principle: the connections between users and sessions (where a user may only own one session at any one time) and between activities and classifications (where each activity is associated with a classification), which are distinct inside the model.

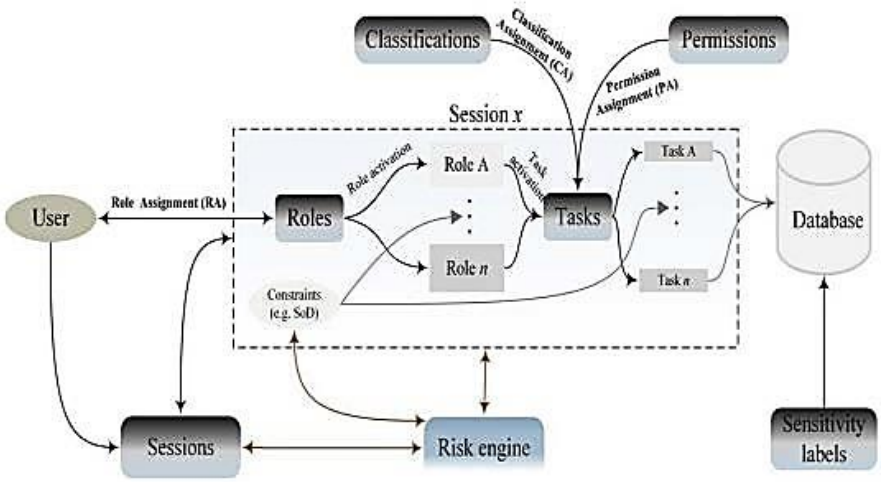

Figure 4; Access control for cloud computing (AC3) (Level 2&3).

3. Hazardous conditions and untrustworthy procedures

The system employs a standard like to that shown in Figure 4, whereby each instance of accessing data or resources is allocated a security label to prevent the potential for security label reuse or fraudulent activities by actions or procedures. To ensure functionality, it is necessary to generate a distinct security token and use it exclusively for each access point or process linked to it.

Data analysis involves the examination and interpretation of data.

While the suggested model is derived from the T-RBAC model, which is desirable because of its simplicity and flexibility, it has to be improved and implemented for distributed computing in order to facilitate the assignment of dynamic and arbitrary standards. Customer behaviour and access on both local and global levels. When building an access control framework, it is important to consider the varying degrees of general knowledge that information might have. The MAC access control framework, which is extensively used, incorporates information responsiveness as a criterion for granting or denying access. However, shipping is a costly and time-consuming procedure. In addition, there exists a significant disparity between the web applications used in the application tiers and the lower tiers. This is due to the fact that the framework components and loops in the lower tiers are regarded as completely reliable. Ultimately, it is unwise to place absolute reliance in them. Due to the continual changes in the connections between clients and assets, achieving the dynamics of distributed computing is inherently challenging. Cooperatives and specialised clientele are most probable to be located in separate security zones. Moreover, as clients are not limited by temporal or spatial restrictions,

the dynamic and capricious behaviours they exhibit pose a significant concern for designers of access control systems.

4. Conclusion

This study introduces a novel access control technique for distributed computing that is both economical and effective. The suggested model can fulfil the input control needs in distributed computing and should be implemented. It collaborates with activity and career prerequisites to expedite and simplify the award procedure. Additionally, it employs the use of transition (macroscopic) and nomenclature prerequisites and demands. Customers are allocated security zones that are suitable for their lawful use and placement inside our premises. Every job in the model is given crucial duties that enable it to carry out its activities effectively. To access a continuous flow of information for the model, the information might be marked with security markers that indicate its sensitivity. Every run is accompanied with a security order that provides instructions on how to access the information or resources required for the task, as well as the specific permissions necessary to successfully carry it out. Therefore, every execution or loop that tries to access the information must be in a sequence that supersedes the information security flags of the chosen resource or its equivalent. A gaming engine is used to regulate the actions of customers that exhibit diverse and unpredictable behavioural patterns. Allocate credit to clients depending on their conduct in certain circumstances. A security label engine may be used to distribute security labels in scenarios that are partially or not completely dependable, depending on the circumstances. Security labels are used in certain situations to convey the degree of trust and security in a given region. The suggested security tag includes just the essential information required for ensuring access security, while still being lightweight. As part of this project, we will create a validation tool that can effectively manage substantial quantities of time and intricate spatial configurations. In addition, we will activate the betting machine and its components, which have the function of regulating the dynamic modes of operation. Once the validation component and the random engine have been completed, the model is deployed and assessed.

References

1. Bhatt, S., Patwa, F., & Sandhu, R. (2017). An attribute-based access control extension for OpenStack and its enforcement utilizing the policy machine. *2016 IEEE 2nd International Conference on Collaboration and Internet Computing*.

2. Al Amri, S. M. S. (2018). *IaaS cloud security enhancement: An intelligent attribute-based access control framework.*

3. Younis, Y. A., Kifayat, K., & Merabti, M. (2017). An access control model for cloud computing. *School of Computing and Mathematical Sciences, Liverpool John Moores University*; Ruj, S. (2017). Attribute-based access control in clouds: A survey. *R.C. Bose Center for Cryptology and Security, Indian Statistical Institute, Kolkata*.

4. Bringer, J., Gallego, B., Karame, G., Kohler, M., Louridas, P., Onen, M., Ritzdorf, H., Sorniotti, A., & Vallejo, D. (2015). TREDISEC: Trust-aware reliable and distributed information security in the cloud. In *EDemocracy Citizen Rights in the World of the New Computing Paradigms* (pp. 193-197). Springer International Publishing.

5. Nguyen, D. (2014). *Provenance-based access control models* (Ph.D. thesis). The University of Texas at San Antonio.

6. Tang, B. (2014). *Multi-tenant access control for cloud services* (Ph.D. thesis). The University of Texas at San Antonio.

7. Younis, Y. A., Kifayat, K., & Merabti, M. (2014). An access control model for cloud computing. *Journal of Information Security and Applications, 19*(1), 45-60.

8. Anggorojati, B., Prasad, N. R., & Prasad, R. (2014). Secure capability-based access control in the M2M local cloud platform. In *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronics Systems (VITAE)* (pp. 1-5). IEEE.

9. Li, B., Li, J., Liu, L., & Zhou, C. (2015). Toward a flexible and fine-grained access control framework for infrastructure as a service cloud. *Security and Communication Networks*.

10. Li, F. (2015). Context-aware attribute-based techniques for data security and access control in mobile cloud environment. *April 2015*.

11. Zhang, Y., Patwa, F., Sandhu, R., & Tang, B. (2015). Hierarchical secure information and resource sharing in OpenStack community cloud. In *2015 IEEE International Conference on Information Reuse and Integration* (pp. 419-426). IEEE.

12. Bijon, K., Krishnan, R., & Sandhu, R. (2015). Virtual resource orchestration constraints in cloud infrastructure as a service. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy - CODASPY '15* (pp. 183-194). ACM Press.

13. Ngo, C., Demchenko, Y., & de Laat, C. (2015). Multitenant attribute-based access control for cloud infrastructure services. *Journal of Information Security and Applications, 27*, 65-84.

14. Singh, D., et al. (n.d.). *International Journal of Computers*. Retrieved from http://www.iaras.org/iaras/journals/ijc