RISK AWARE HIERARCHICAL ATTRIBUTE SET-BASED ENCRYPTION (RA-HASBE) ACCESS CONTROL MODEL

RajaniKanth Aluvalu [1], Lakshmi Muddana [2]

[1] **Department of CSE, Vardhaman College of Engineering, Hyderabad, India**
[2] **Department of Information Technology, GITAM University, Hyderabad, India**

Corresponding author: RajaniKanth Aluvalu, rajanik.rkcet@gmail.com

ABSTRACT: Business organizations widely accepting cloud computing to handle their complex business process and increased business transactions. Organizations IT infrastructure and IT management had moved onto the cloud infrastructure and accessed through third party network. Cloud computing delivers on-demand services over the internet. Cloud service provider must ensure the security of the data and processes to the cloud users. Multitenancy being one of the key features of the public cloud. It is required to entrust security and privacy of the users outsourced valuable data. Several access control models have been proposed for cloud computing. Being highly dynamic cloud computing environment demands flexible, fine-grained, dynamic access control models. Business processes are expecting on time data for their analysis and client needs. Encryption based access control models proved better for cloud computing to hold outsourced data. We can prevent access to the encrypted data by hiding keys to decrypt the data. Cloud computing storage being remote to the user demands encryption-based access control models. We require access models with dynamic policies and with dynamic authorization. In this paper, we will be discussing the design, framework, and development of RA-HASBE access control model having dynamic authorization mechanism. We will explore the implementation and analysis of RA-HASBE access control model.

KEYWORDS: Access control, Risk, Dynamic policy, Dynamic authorization, Attribute, Cloud computing, Encryption, Decryption, Data Security.

1. INTRODUCTION

Cloud computing has speedily become a widely accepted and spread computing model. The success of cloud computing depends on the internet connectivity and security provided to the outsourced data. Cloud computing being widely distributed delivers services over the internet. Cloud service provider (CSP) is responsible for managing a cloud environment. CSP has to ensure trust and security of the data stored in large amount by the data owner on the cloud. Confidentiality of stored data can be protected by proving access control models as an authorization mechanism. An access control model helps us in restricting unauthorized access to sensitive data by users (Aluvalu et al., [KA15]). Accessing means able to add delete and append the data. Permission to access a resource is called authorization. Earlier various traditional access control models like DAC (Discretionary Access Control), MAC (Mandatory Access Control), and RBAC (Role based access control) (Aluvalu et al., [KA15]). They are not sufficient for providing security to data in cloud computing environment. Later Attribute-based Encryption Schemes are proposed for providing security to outsourced data.

Attribute-Based-Encryption (ABE) model, Sahai and Waters proposed in the year 2005. In ABE data is encrypted and decrypted using user attributes. User's secret key and the cipher text are dependent upon attributes. To decrypt user's data, ciphertext attributes should match with attributes of the user key. The major disadvantage with attribute-based encryption (ABE) scheme is that data owner needs to use the public key of every authorized user to encrypt data. ABE demands both data owner and consumer to be online for exchanging keys. Later various ABE-based access control schemes have been proposed to overcome this problem. We will be discussing few of them.

(KP-ABE) was proposed by Goyal et al in 2006 [G+06]. In this model cipher text is associated with a set of user attributes and the private key is associated with access structure. The user can decrypt the cipher text only when the associated attribute set satisfies the access structure. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) was proposed by Bethencourt et al., [BSW07]. In this model cipher text is associated with access structure and the private key is associated with a set of user attributes. The user can decrypt the cipher text only when the user attributes satisfies the access structure associated with the ciphertext.

Hierarchical Attribute Set-Based Encryption (HASBE): This access control is a combination of HIBE and CP-ABE. In HASBE users are arranged in hierarchical order otherwise, it is same as CP-ABE.

User hierarchical order is based on their roles and designations. Higher order roles overlap lower order roles (Kamliya et al., [KA15]).
The root master will be on the top followed by domain masters. Domain masters consist of user sets. Access control model is highly scalable because of its hierarchical structure. Like CP-ABE and KP-ABE also stores data in encrypted format on the untrusted server. However still, HASBE suffered from various drawbacks like handling compound attributes, lack of flexibility in the authorization, lack of efficient key management mechanism.
HASBE (Hierarchical Attribute Set-Based Encryption) is extended for supporting sub-domain level hierarchy. The extended model supports secure key distribution to access the files that are stored on cloud-based on roles. In the extended model, it is not required for the data owner to be always online. Key distribution will be handled by trusted authority (TA) (always online) in the more secure way. Data owner will share keys and specific role based policy with trusted authority. TA will distribute keys to data consumers on request if they satisfy the data owner's predefined policy. HASBE will maintain user-level domain hierarchy using user attributes. Creating sub-domains within user domains will improve the system performance.
We can reduce the burden of handling increasing user requests on domains by creating sub-domains. This makes HASBE system highly scalable in terms of increased user registrations.

2. EXISTING SYSTEM

As mentioned in the below Fig. 1 HASBE access control model holds the following major functionalities: Trusted authority, Domain Authority, Data Owner, Data Consumer. The user stores data on the cloud in encrypted form. Data consumer by satisfying the access policy using his attributes can access and decrypt the data using the private key provided. This mechanism helps us in keeping the data confidential.
As shown in figure 1 higher level authority will authorize lower level authorities. The biggest issue with cloud computing is a loss of data ownership. To secure the confidential data from unauthorized access data owner will store the data in encrypted form on cloud and will generate a secret key for each individual file.Data consumer upon request will get the take that key from domain authority/trusted authority, by using which data consumer will decrypt the data. The entire hierarchy of the system is as shown in figure 1.

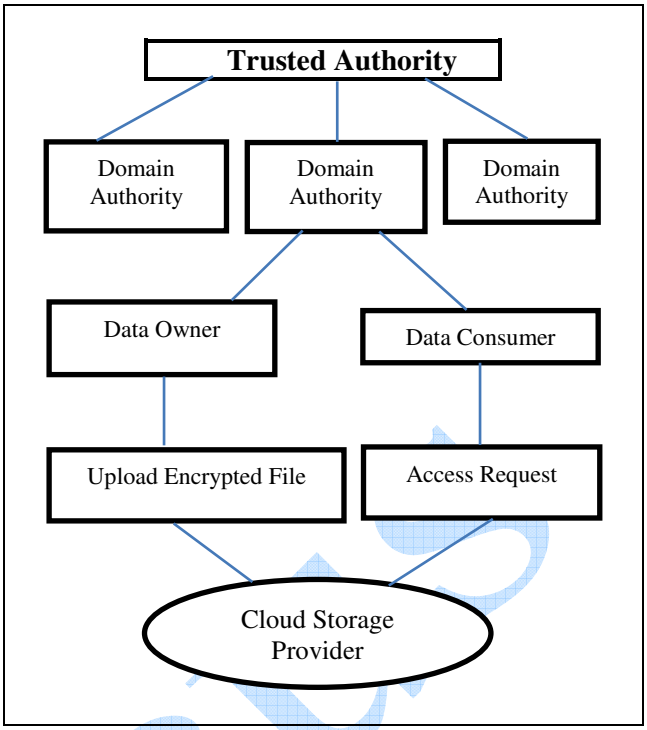

Figure 1. HASBE Architecture

The user has to register with the system by using his user attributes. Once the user submits his request the higher authority will approve user and will provide key to be used by the user at the time of login. Now the registered user can store data on the cloud in encrypted form and will define access policy. The user can access and decrypt the data stored on the cloud by satisfying the access the policy defined by the data owner. In the absence of a lower level user, the higher level user can access lower level user data by using master key and is responsible for all work related to lower authority.
In the existing model, if the higher level authority is not available, the complete organization work will be kept on hold waiting for the permissions. Higher authority will have access to privileges of lower level employees, whereas vice versa is not allowed. At times lower level authorities have to perform the roles of higher level in their absence to smoothly complete business transactions.

Here:
D = {D1, D2, D3, D4}
Where,
D is Cloud
D1 is CEO.
D2 is General Manager.
D3 is the list of managers.
D4 is the list of employees.

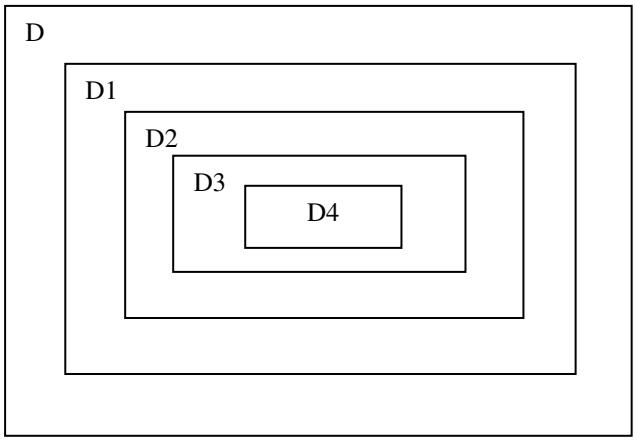

Figure 2. Domain Hierarchy (Kamliya et al., [KA15])

3. PROBLEM DEFINITION

Cloud's dynamic nature and organizational on demand work requires a Risk-aware role based flexible access control models with encryption. Data owners are facing a serious threat from unauthorized access of their confidential data; even sometimes due to lack of physical control and efficient security mechanisms on outsourced data they are missing their data. Access control mechanism with efficient access control models has to overcome the said security threat. The organizational demand of completing transactions on time requires low-level employees performing responsibilities of the higher authorities in their absence. However, traditional access control mechanisms are based on static policies which make them too rigid to handle the complex situations.
The data owner defines a privacy policy for every data file stored in the cloud. He wants to prevent the unauthorized access to his file and holds the access control on his file with himself, without relying on CSP. The major features of our proposed model include:

- *Fine-grained access control:* Different users are permitted to read different sets of data based on satisfying data owner access policy.
- *User revocation:* model allows us for revoking User's access privileges to restrict from future access of data.
- *Flexible policy specification*: Model allows us to create complex access policies in a flexible manner.
- *Scalability:* Model should handle increased number of users and efficiently handle user management, storage, and Key management.
- *Dynamic access*: to ensure that users on emergency are allowed to access restricted files through risk computation.
- *Privacy-preserving:* The system should ensure that user's data privacy is maintained, even for TPA.

We addressed the above issues by allowing data owner to encrypt the data and define access policies. We have developed a mechanism using user attributes to evaluate the risk of allowing access to user failed to satisfy the access policy defined by the data owner to make the system more dynamic. Data owner is allowed to perform the computation tasks with fine- grained data access control on data stored on remote servers like a cloud (Zissis et al., [ZL12]).
We named our model as Risk-aware access control model (RA-HASBE). It is a combination of HASBE and RAAC access control models. HASBE is an extended access control model from CP-ASBE providing a hierarchical set of users providing scalable, flexible and fine-grained access control (Bijon et al ., [BKS13]).

4. IMPLEMENTATION MODEL

Implementation of the proposed work is done in two major steps
1. Developing strategy for Risk calculation and designing risk engine,
2. Integrating Risk engine with Enhanced HASBE. The architecture of implemented model is given below.
Major Functionalities of proposed Scheme:
System Setup, TA Grant, DA Grant, New Domain Administrator/User Grant, Risk Engine setup, New File Creation.

Part A: Risk-aware access control model:

Anywhere computing demands sharing of information in dynamic computing environments using the third party network with user hierarchies vice versa to complete the business transactions on time. This had created a requirement for risk-aware access control systems will help us to share information securely among authorized users by assessing the risk (Kamliya et al., [KA15]).
All such dynamic business models outsourced their data on cloud servers. The standard access control models (Role based, attribute based) are suitable to operate in the stable environment and do not evaluate the risk of allowing access to the data. Risk-aware access control models differ from the other access control models discussed; 'Risk' is the key metric, considered for taking decisions on data sharing (Sood, [Soo12]; Bijon et al., [BKD13]; Diep et al., [D+07]).
The major objective of risk-aware access control (RAAC) systems is to provide a mechanism that can manage the trade-off between the risks of permitting unauthorized access (Data and files of higher level employees) with the cost of denying access for not satisfying the access policy and the inability to access resources may have profound consequences (Kandala et all., [KSB11]).

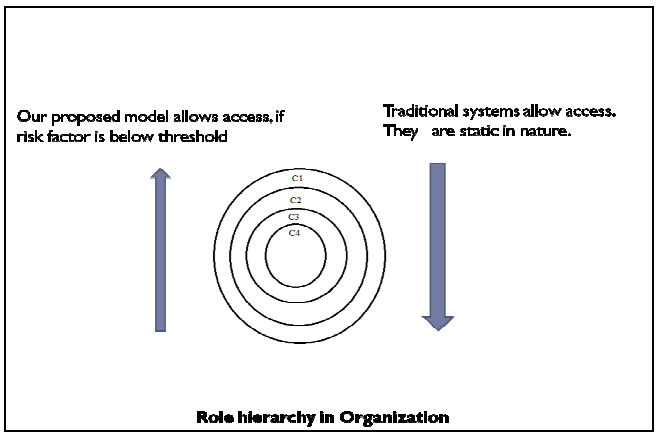

Figure 3. Role hierarchy in Organization

Figure 3 represents the difference between our proposed model and traditional model. Generally, in role based access control models, employees in the higher hierarchy are able to access the data of lower hierarchy employees. The beauty of our system is it also allows employees in the lower hierarchy to access the data of employees in higher hierarchy temporarily. This approach is particularly useful to take access control decisions dynamically in an emergency situation (Molloy et al., [M+12]). An employee can get the privileges of accessing the data for the session by assessing the risk.

Figure 4 represents risk assessment process in our scheme. When the user fails to satisfy the access policy to access file. He can request Risk engine for access, upon receiving request risk engine will assess the user attributes (Cheng et al., [C+07]; Ni et al., [NBL10]).

X = set of attributes that satisfy the user policy

Y = set of attributes that do not satisfy user policy.

P = Primary attribute specified by the user during defining policy.

The user will be granted access by risk engine if P= true AND X > Y. i.e requested user attributes satisfy primary attribute and number of attributes that satisfy policy are more than attributes that do not satisfy the policy. Otherwise, grant request denied (Chen et al., [CGN13]).

PartB: Hierarchical attribute set based encryption (HASBE) Access control Model:

We have integrated risk engine with HASBE. Figure-5 represents proposed RA-HASBE architecture. The security of HASBE is equally proved with CP-ASBE. HASBE varies from CP-ASBE in the hierarchical structure of users.

Data consumer will request domain authority/trusted authority for file access. If data consumer fails to satisfy user-defined access policy, access to the file will be denied. In an emergency, the user can send a request to risk engine. Upon receiving request risk engine will assess the requested user and will grant access if he//she satisfy the risk strategy, otherwise denied. RA-HASBE is very much dynamic in terms of granting access.

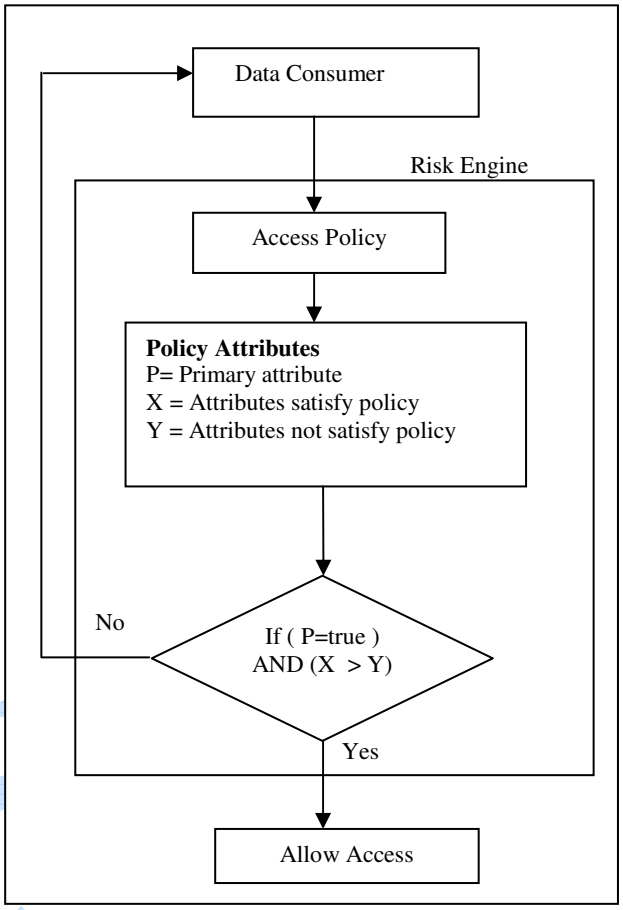

Figure 4: Risk Assessment Process

5. RESULTS

We will first evaluate the computation complexity for the system setup and then evaluate each critical functionality of RA-HASBE.

System Setup: - Time taken for the system setup is linearly proportional to the number of attributes. Increased attributes will increase the time taken for system setup. The setup operation completes in a fixed time. The computation complexity of Setup operation is O(1).

Top-Level Domain Authority Grant:- Here TA (Trusted Authority) will create a new user or domain authority. The MK("Master Key") of a DA("Domain Authority") is in the form of MKi= ("A,D, D_{ij}, D_{ii} for all a_{ij} belongs to A, E_i for A_i belongs to A") where "A" is a key structure allied with a "New Domain authority", Ai is the set of A . Let N be the number of attributes in A, and M be the number of groups in A. Two exponentiations for each attribute in "A" and one exponentiation for each group in A are required for the computation of MK_i. The computation intricacy of "Top-Level Domain Authority" grant operation is O (2N+M).

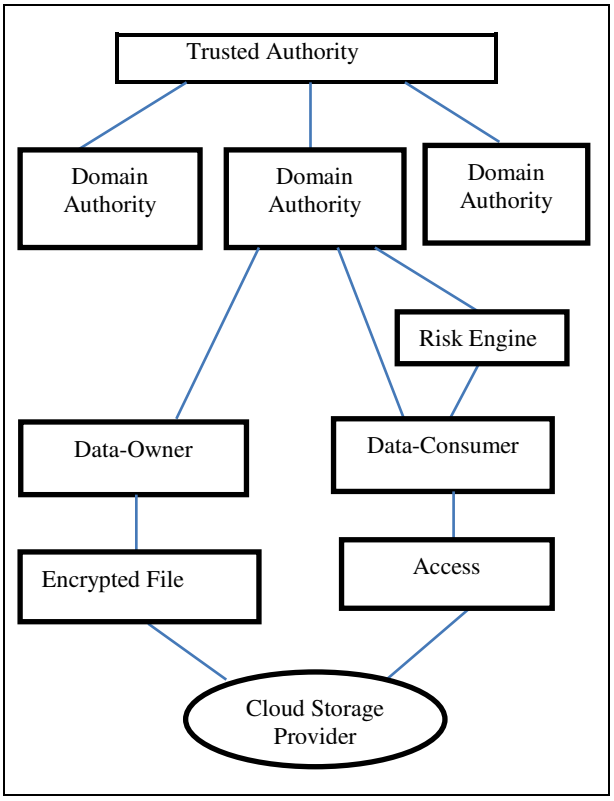

Figure 5. RA-HASBE Architecture

Sub-Domain Creation: - Similar to DA, process is executed by the TA and the MK of sub-domain is in the form of MKi= ("A, D, D_{ij}, D_{ii}, for a_{ij} belongs to A, E_i for A_i belongs to A") where "A" is a key structure allied with a "New Domain authority", A_i represents the set of A. Sub-domain creation do not increase any kind of complexity.sub domain uses the same keys generated by top level domain hierarchy. We will be clustering the users of the domain within the sub-domain. Keys are allocated to only parent's domain. Hence computation complexity of Sub domain authority is O(2N+M).

New User/Domain Authority Grant:- New user will be associated with sub-domain and top level domain authority. A new user has to provide his values for the attributes defined by the domain authority for registration. With the new user registration, the keys are required to be re-randomized, the computation complexity is O (2N+M). Where N is the number of attributes in the set of the new user or domain authority, and M is the number of sets in A.

New File storing: - The user needs to encrypt data file using the Blowfish algorithm during file creation. The size of the data file affects the time taken by blowfish algorithm to encrypt. Blowfish uses two exponentiations for encryption, for every foliage lump in T and one exponentiation for every interpreting lump in T. The Computation Complexity of new file storing is ("2|Y|+|X|"). The below table 1 summarizes the computational complexity of the enhanced HASBE with existing model.

Table 1. Computation Complexity of RA-HASBE and HASBE

Operations	RA- HASBE	HASBE
System setup	O(1)	O(|Y|)
Top-Level-DA Grant	O(2N+M)	NA
User/DA Grant	O(2N+M)	O(|Y|)
Sub-Domain Grant	O(2N+M)	NA
User sets	O(1)	NA
File Creation	O(2|Y|+|X|)	O(|1|)
File Deletion	O(1)	O(1)
User Revocation	O(1)	O(1)
Risk Engine	O(1)	NA

User Revocation: Domain authority will perform user revocation. This operation requires fixed amount of time and the time complexity of this process is O(1).
RA-HASBE grants permission to access files by the user when the risk factor is below threshold else the access grant is denied. Below table summarizes the functionality of RA-HASBE.

Table 2. RA-HASBE Comparison with existing system

Operations	RA- HASBE	HASBE
Access grant to file with risk factor below threshold	Access granted	Access not granted
Access grant to file with risk factor above threshold	Access not Granted	Access not granted

Table 3 shows the experimental results of HASBE and RA-HASBE. The first row in the table shows with the same set of attribute values data consumer is able to get access to data using RA-HASBE ensuring the security of data, whereas access request is denied in HASBE model. This proves RA-HASBE is dynamic in nature.

Table 3. Experimental Comparison

Access policy	Data Consumer attributes	HASBE	RA-HASBE
Deptid= prodn AND loc=surat AND designation=manager.Prime Attrib=Dept id	Deptid=prodn AND loc = surat AND Designation=clerk	Access denied	Access granted
Deptid= prod AND loc=surat AND designation=manager.Prime Attrib=Dept id	Deptid=mktng AND loc = surat AND designation=manager	Access denied	Access denied
Deptid= prod AND loc=surat AND designation=manager.Prime Attrib=Dept id	Deptid=prodn AND loc=mumbai AND designation=clerk	Access denied	Access denied

6. CONCLUSION

In this paper, we had discussed the implementation of our RA-HASBE access control model which is highly dynamic in terms of allowing access to the users. This model is highly scalable, flexible in terms of user access management. Our model is best useful for organizations with hierarchical roles and encounters emergency works. In future, we want to extend our scheme for handling public auditing.

REFERENCES

[BKS13] **K. Z. Bijon, R. Krishnan, R. Sandhu -** *Towards an attribute basedconstraints specification language*. In Social Computing (SocialCom), 2013 International Conference on (pp. 108-113). IEEE, 2013.

[BSW07] **J. Bethencourt, A. Sahai, B. Waters -** *Ciphertext-policy attribute-based encryption.* In 2007 IEEE symposium on security and privacy (SP'07) (pp. 321-334). IEEE, May 2007.

[CC11] **L. Chen, J. Crampton -** *Risk-aware role-based access control*. In International Workshop on Security and Trust Management (pp. 140-156). Springer Berlin Heidelberg, 2011.

[CGN13] **L. Chen, L. Gasparini, T. J. Norman -** *XACML and risk-aware access control.* Resource, 2(10), pp.3-5, 2013.

[C+07] **P. C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, A. S. Reninger -** *Fuzzy multi-level security: An experiment on quantified risk-adaptive access control.* In 2007 IEEE Symposium on Security and Privacy (SP'07) (pp. 222-230). IEEE, 2007.

[D+07] **N. N. Diep, L. X. Hung, Y. Zhung, S. Lee, Y. K. Lee, H. Lee -** *Enforcing access control using risk assessment.* In Universal Multiservice Networks, 2007. ECUMN'07. Fourth European Conference on (pp. 419-424). IEEE, 2007.

[G+96] **V. Goyal, O. Pandey, A. Sahai, B. Waters -** *Attribute-based encryption for fine-grained access control of encrypted data.* In Proceedings of the 13th ACM conference on Computer and communications security (pp. 89-98). ACM, October 2006.

[KA15] **V. Kamliya, R. Aluvalu -** *A Survey on Hierarchical Attribute Set based Encryption (HASBE) Access Control Model for Cloud Computing*. International Journal of Computer Applications, 112(7), 2015.

[KSB11] **S. Kandala, R. Sandhu, V. Bhamidipati -** *An attribute based frameworkfor risk-adaptive access control models.* In Availability, Reliability and Security (ARES), Sixth International Conference on (pp. 236-241). IEEE, 2011.

[M+12] **I. Molloy, L. Dickens, C. Morisset, P. C. Cheng, J. Lobo, A. Russo -** *Risk-based security decisions under uncertainty.* In Proceedings of the second ACM conference on Data and Application Security and Privacy (pp. 157-168). ACM, 2012.

[NBL10] **Q. Ni, E. Bertino, J. Lobo -** *Risk-based access control systems built onfuzzy inferences*. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (pp. 250-260). ACM, 2010.

[RL15] **R. Aluvalu, L. Muddana -** *A Survey on Access Control Models in CloudComputing.* In Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1(pp. 653-664). Springer International Publishing, 2015.

[San88] **R. Sandhu -** *Transaction control expressions for separation of duties*. In Aerospace Computer Security Applications Conference, 1988., Fourth (pp. 282-286). IEEE, 1988.

[Soo12] **S. K. Sood -** *A combined approach to ensure data security in cloudcomputing*. Journal of Network and Computer Applications, 35(6), pp.1831-1838, 2012.

[VG12] **V. Venkatakrishnan, D. Goswami (eds.) -** *Information Systems Security*. 8th International Conference, ICISS 2012, Guwahati, India, Proceedings (Vol. 7671). Springer Science & Business Media, December 15-19, 2012.

[W+11] **I. Wakeman, E. Gudes, C. D. Jensen, J. Crampton (eds.) -** *Trust Management* V:5th IFIP WG 11.11 International Conference, IFIPTM 2011, Copenhagen, Denmark, Proceedings (Vol. 358). Springer Science & Business Media, 2011.

[ZL12] **D. Zissis, D. Lekkas -** *Addressing cloud computing security issues*. Future Generation computer systems, 28(3), pp.583-592, 2012.