International Journal of Engineering & Technology, 7 (3) (2018) 1795-1800

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i3.13228
Research paper

A Novel access control model in cloud computing environment (PAR-AC)

Rini Mahajan [1] *, Manish Mahajan [2], Dheerendra Singh [3]

[1] Ph.D. Research Scholar I. K. Gujral PTU Jalandhar, Punjab
[2] Professor, Department of Computer Science & Engineering, Chandigarh Engineering College, Landan, Mohali, Punjab
[3] Associate Professor, Department of Computer Science & Engineering, CCET, Sector 26,, Chandigarh
**Corresponding author E-mail: rinimahajan@gmail.com*

Abstract

Cloud computing has changed the perspective of information storage worldwide; still it has not fully matured. Security, privacy and access control of the stored data is the biggest concern in the IT industries and these are the major research areas in the field of cloud computing. Although many researchers have proposed variety of access control methods, still there is a scope of improvement. The main objective of this paper is to study the limitations of formerly proposed access control methods and to propose a better new access control method. It is the combination of Attribute based access control (ABAC), Hierarchical Role based access control (HRBAC) and Policy based authentica-tion. As a result we got a fine grained access control method. It also includes the feature of role delegation which is very important in pre-sent scenarios. Comparative analysis shows that proposed method combined many new features with existing access control methods to make the system sound.

Keywords: *Cloud Security; Access Control; Scalability; Policy Based Authentication; Two Factor Authentication.*

1. Introduction

Cloud computing comes with many features such as flexibility, elasticity, scalability, universal access, economic etc. These features make accessibility of data much easier and reduce the headache of carrying physical devices such as hard disk for carrying data everywhere. but this feature makes our data on risk for various threats and attacks[1][2]. Due to remote storage, owner is too much concerned about the loss of his data[3] .One of the ways is to secure the data with authentication where user has to login before using cloud services. However, authentication is not strong enough as after authentication, data whether it is normal or sensitive, will be available to all authenticated cloud users. Organizations have a huge amount of data with different sensitivity and privacy level, so it should not be available to everyone. There must be implementation of authorization with authentication [4] [5]. Although a lot efforts have been put in this field to provide integrity, security, and availability in cloud environment, despite of all the efforts we cannot say that cloud environment is fool proof. It is a common belief that the trust in today's cloud application is not sufficient [6].

Choosing a proper access control method is a vast research topic due to enormous development in distributed network [7]. The number of users is not limited in cloud environment; there can be cases of multiple accesses of same data by multiple users. In order to protect the data and to maintain trustworthiness in the cloud environment, an efficient access control mechanism should be deployed [8][9].Some traditional access control methods are User based access control method (UBAC), Mandatory Access Control (MAC), Discretionary Access Control (DAC) and Role Based Access Control (RBAC) [10] [11] [12] [13]. These models can be good enough for standalone machines but not efficient for cloud environment where large servers are involved with large number of users. For that reason researchers worked out to find suitable methods for cloud environment. It is shown in various studies that traditional RBAC method is a robust model and can be extended to distributed environment[14]. Further, three versions of RBAC were proposed: flat RBAC, constrained RBAC, hierarchical role based access control model (Hierarchical RBAC)[15] . Flat RBAC is the simplest form of RBAC and is useful when limited number of users is there i.e. for small-scale organization. To define the hierarchy of roles, Hierarchical RBAC is defined [16]. However, this model provides unlimited access to the users as the user might have more than one role. To solve a issue constraint based RBAC was designed in which additional check constraints were added to hierarchy of roles. From all above discussed methods hierarchical RBAC has been deployed in cloud environment with the problem of unlimited access [15] . An ABAC mechanism is also found to be good for cloud environments [17] [18]. If we consider ABAC and RBAC individually then they shows some limitations in cloud environment [19].

RBAC does not mention how many users per role are allowed. This makes the system prone to hacking. In this model granularity of authorization is very large. It does not keep back up copy of data. It sends data and information straight to the server. If administrator makes some mistakes then it may lead data to incorrect security format and also if in any session data get missed it is not available by back up rather we have to run a query again. In RBAC it is difficult to decide right role for each subject. Roles are decided on the basis of attributes. For n attributes, 2^n roles can be defined. With the number of attributes, number of roles will also be increased. This results in low efficiency. It does not provide flexible access control policy[7]. Some of the works has also been done on ABAC alone in clouds[20] [21] [22],[23] and [24]. All these research works are based on attribute based encryption ABE

Copyright © 2018 Rini Mahajan et. al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

[2]. If ABAC deployed alone then there might be a case of attribute duplicity. It may happen that two requestors are having similar subject, object and environmental attributes. As size of cloud is very big, so to maintain truthfulness of stored data through ABAC access control model becomes less effective. Hence, in that case one can merge role based access control with ABAC approach. Further, A hybrid approach is proposed, which combine ABAC with RBAC with defined hierarchy [7]. But this model does not solve the problem of back up of data.

Further, cloud optimized RBAC had been designed with [3] level of authentication. Due to this response time and complexity of system increased. This method could not resolve the problem of unlimited access of data [25]. Then trust based evaluation to flat role based ACL approach was added [26]. This model has static evaluation of trust. Therefore, if trust verifier entity is of malicious nature then it will let down the whole TRBAC scheme. Later on access control for cloud computing (AC3) was proposed [27], which defines various sensitivity levels for stored information. Every role is assigned a set of tasks and every task is assigned a security label for accessing data/information. This model does not solve the problem of back up of data. Next Profile based access control in cloud computing environments [28] and a dynamic risk-based access control model for cloud computing [29] were designed by researchers. But these models does not consider authentication with authorization.

This paper considers the problem of back up of data, unlimited access of data, linking of authentication with authorization, security of stored data and proposes a new novel access control method.

2. Novel access control method (PAR-AC)

Health care is one of the main areas where access control is very important. Tremendous records related to patients are stored on cloud; this information is very personal and sensitive. Proposed work is practically carried out on a module of a healthcare system. Hospital management system has been deployed on cloud with proposed access control method PAR-AC (Policy based, Attribute based, Role based, Access control). It helps to manage and utilize resources efficiently. To understand the PAR-AC, consider the following scenario:

If one of the doctor member x have registered a patient and examined him and uploaded description of disease uploaded on a cloud environment. In addition to that consider a higher role is assigned to one of the doctor as head of the department (HOD). In Hierarchical RBAC, HOD doctor will be able to see all the information of the patients uploaded by the junior doctors. i.e. This model provides unlimited access to root level role [16]. PAR-AC attaches different parameters with role hierarchy policy to grant access and turns the access of the data in to limited access. If we attach disease/department domain attribute with patient, doctor and HOD then situation will be in control. If a person having HOD's role not having attributes similar to the attributes attached with the patient and doctor x, he would never going to access the description and medical information of patient written by doctor x. In this way attributes are combined with role in PAR-AC. While deciding the role we can divide attributes in to two parts: fixed attributes and variable attributes. On the basis of fixed attributes roles of users are decided. With variable attribute additional dynamic decisions are taken. This combination of role based and attributes based access method increases performance in cloud environment. Moreover PAR-AC links access control mechanism with authentication. Here type of authentication is based on privilege level of the user. If we implement high level security system for all type of users then the system will become very complex and efficiency will degrade. To implement this factor we have proposed policy based authentication. The security tags and sensitivity labels are assigned to various stored data and permissions. Data is present in encrypted format in the database.

2.1. PAR-AC contains following steps

1) Registration of new user
2) for existing user login (policy based authentication)
3) After successful authentication –Access Control Mechanism is applied which allows user to interact with the system based on authorization level or privileges allocated to him.

In following section all the steps of PAR-AC are described in detail

2.2. Registration of new user

For this module, following are the steps

1) User browse and request for specific service from the cloud by enters URL of application.
2) The cloud system response and sends registration form to the user.
3) Upon receiving the registration form, the user fills and submits it to the cloud.
4) After submitting registration form the Cloud checks and processes the registration from. If the details entered by the user are correct and satisfied to the condition terms a link is sent to user's email.
5) The user login into his mail and follows the link received from cloud,
6) After confirmation user becomes the registered user and user is having credentials for login.
7) After successful registration process Admin decides the type of user and assign the privilege level. Further, it helps in fixing up the type of authentication required as shown in Fig. 1.

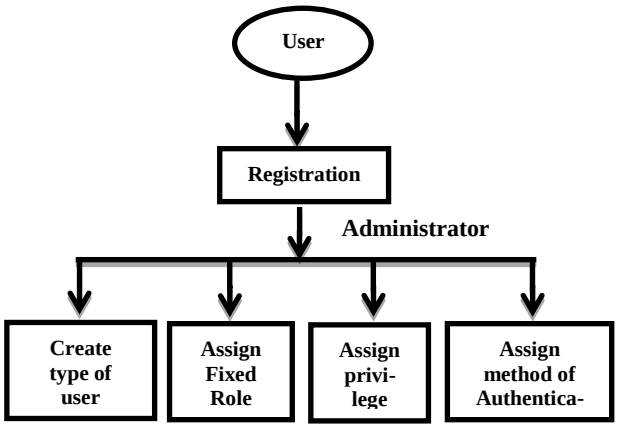


Fig. 1: Admin Jobs after Registration in PAR-AC.

2.3. Policy based authentication- (Single sign on)

Administrator passes through three factor authentication as he has total control over the system. For other users following steps are used:

1) User fills user name and password and submit it.
2) System checks the type of user and its privilege level.
3) If user belongs to low privilege level then user gets authenticated, if he has provided correct user name and password.
4) If user is having high privilege level then Two-factor authentication will be applied.
5) One code is sent to user's email.
6) User has to fill that code on this application.
7) If code is correct then he will login successfully.
8) Once user is authenticated he can use all the facilities. User is not required to login again and again for various services. This is known as the facility of single sign on which is incorporated in this proposed system as shown is Fig. 2.

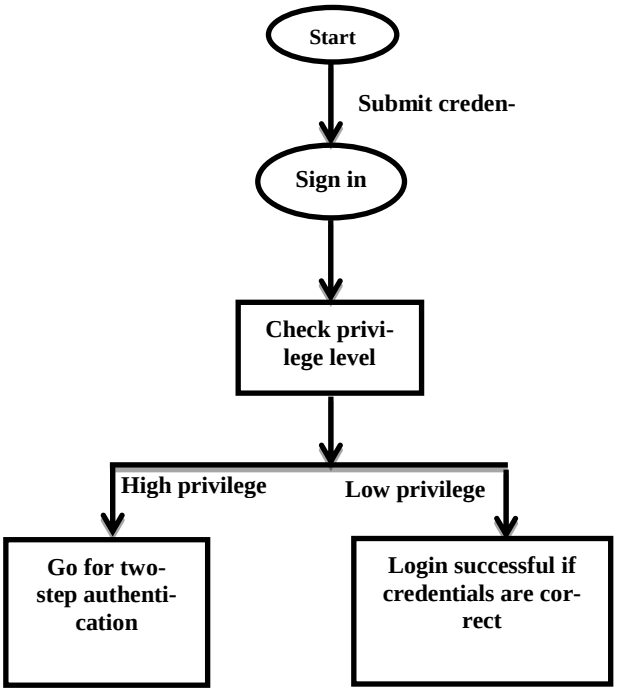


Fig. 2: Policy Based Authentication in PAR-AC

2.4. Access control algorithm

It is shown in Fig. 3.

1) Check user role algorithm
2) Check Resource and user attributes algorithm.
3) Evaluate and take decision for granting access.

2.5. Back up algorithm

1) 1 Whenever authenticated and authorized user sends a query then result is sent to user. At the same time back up of result is created to support recovery of data loss.
2) The backup will be store for a particular session.
3) When the session will be over data will be erased and memory will be free. So that memory management can be done properly.

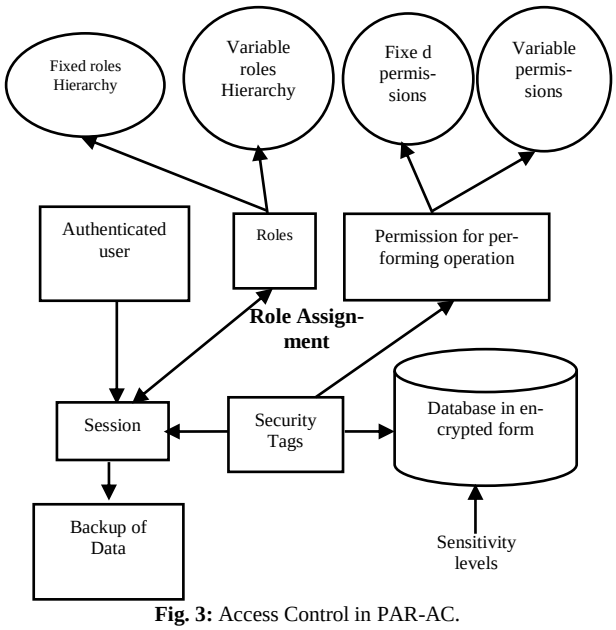


Fig. 3: Access Control in PAR-AC.

3. The PAR-AC model structure

The essential idea of PAR-AC model is that roles are a collection of fixed attributes, while dynamic nature of role is based on frequently varying attributes. The Components of the PAR-AC model are as follows:

3.1. Mathematical operations

Let Us be a system S= {….}
Start of the web Server as s S = {s…}
Let s = {Start of the web Server: 1. Log in with Server.2. Deploy the web application on web Server.}
3. Log out system S = {s, e ...}

Let e= {Log out by user or end of session.}

Users (U)
Roles(R)
Permissions (P)
Privilege levels $\{prl_1, prl_2 \ldots prl_k\}$.
Fixed roles FR,
Varying roles (VR)
Fixed role permissions (FP),
Varying role permissions (VP)

Users = $\{u_1, u_2 \ldots UN\}$

Sessions = $\{s_1, s_2 \ldots s_i)$

Roles $\subseteq$ FR X VR ;

Operations = $\{op_1, op_2 \ldots op_k\}$

Object sets = $\{O_1, O_2 \ldots O_n\}$

Permission finalizing set Perms: FP $\cap$ VP

3.2. Model relations and functions

- UA $\subseteq$ Users X R many to many mapping of user to role allocation relation;
- PA $\subseteq$ Perms X Roles , many to many mapping of permission to role allocation relation ;
- Fixed permission allocation FPA$\subseteq$ FP X FR, many to many mapping of FR to FP allocation relation ;
- variable permission allocation VPA$\subseteq$ VP X VR, many to many mapping VP to VR allocation relation ;
- OPA$\subseteq$ OPS X (FOR X VOR) , a many to many mapping of OP to OR assignment relation
- FUA$\subseteq$Users X FR, a many to many mapping of user to FR assignment relation ;
- VUA$\subseteq$Users X VR, a many to many mapping of user to VR assignment relation;
- SL-$\{SL_1$, SL2----$SL_n\}$ sensitivity levels - is a set of sensitivity levels used to restrict access to data according to its sensitivity.
- Security tags (ST) is a set of security tags $\{ST_1, ST_2$----$ST_n\}$;
- Every user u in the model can have an outlined number of roles $\{r_1.r_n\}$, and every role can have a defined privilege level $\{prl_1 \ldots prl_k\}$. Every privilege level is assigned the exact needed permissions $\{p_{1\ldots} p_n\}$ to accomplish its job, and a classification to access the targeted data or assets.
- F_Session (U): maps user to a set of sessions.
- F_SU (S_i): session-> user it is a function which maps a session to a particular user.
- F_SR (S_i): session-> Roles a function, which maps session to a set of roles.

- F_FSR(S_i): session-> SR a function which maps session to a set of SR
- F_VSR (S_i): session-> VR a function that maps session to a set of VR.
- Allocate _ FP (FR): a function, which maps fixed roles on to set of fixed permissions.
- Allocate _ VP (VR): a function, which maps variable roles on to set of variable permissions.
- Allocate _ P (R): a function, which maps roles on to set of permissions.
- Login (username, password): This function is used to send credentials.
- Verify (username, password): This function is used to verify credentials.
- Check_priv (u): This function checks privilege level assign to the user.
- Verify (u): a function, which checks login successful, or not.
- Send (code, email): a function that sends randomly generated code to registered email.
- Submit (u, code): a function that accepts the code submitted by user.
- Verify (u, code): a function that checks whether code is correct or not. If correct, code then login successful.

3.3. Algorithm

1) Start
2) S □□ sessions
3) If (S== null) then return fail and terminate
4) Else user login
5) Login(username ,password)
6) Verify(username, password))
7) Check_priv(u)
a) if user == low_privilege level then login successful if verify(u)=true else fail
b) if user == high_privilege level then
c) Send(code, email)
d) Submit(u, code)
e) If Verify (u, code) == true login successful else fail
8) If login false then return fail and terminate
9) Else extract priv_level={prl1 or prl2…. prln)
10) Assign roles = {R1 OR R2… Rn)
11) r □□ roles
12) FR □□ FRole(s), p_owned_FR =allocate _ FP (FR);
13) VR □□VRole(s), p_owned_VR =allocate _ VP (VR);
14) P_owned = P_owned_FR & & P_owned_VR
15) P_owned_r =allocate _ P (R);
16) Endif
17) If(P_owned_r == P_ owned) && If (P_owned_r_ST == P_ owned_ST)
18) P_allowed=allocate_ P (R);
19) If(P_owned == P_ allowed) return SUCCESS;
20) Else return FAIL;
21) If session==time out then logout

As illustrated in above, in PA-RAC model, we first extract the current sessions of user u, which has been authenticated with policy based authentication. Then the system extracts the privilege level assigned to the user at the time of registration.. Then, from the sessions of the user u, we can get all permissions of the roles in session s, which extract from both fixed role assignments and variable role assignments. After the above steps, we get all access privileges of the user u, and can verify the correctness of the access privileges. If correct, the access control system grants the corresponding permissions to the user, and returns SUCCESS; otherwise, deny the request and return FAIL.

4. Results and discussion

The PAR-AC is using the combination of ABAC and Hierarchical RBAC methods; as a result we got a fine grained access control method. PA-RAC is very much flexible and easily adaptable in cloud environment. In this method users have been assigned fixed roles based on fixed attributes. Many dynamic roles can be assigned to the users based on varying attributes. Separation of duties is based on hierarchy of fixed roles as well as dynamic attributes. For example one fixed role in hospital management module is doctor and one of the dynamic roles is HOD. A doctor can be the HOD of the particular department. HOD Doctor is having privileges of both HOD and doctor and he will be higher in hierarchy. But the access of HOD is limited to the department to which he belongs. He cannot access the data of doctors of other department. Total functionality of system based on role privileges i.e. system follows the principle of least privilege. According to it, no user gets too much authority; otherwise the system will become risky.

PAR-AC is having all basic features of cloud based access control methods as AC3[27] and HA-RAC [7]. It stores Auditing log. These Auditing logs help to keep track of all the activities perform by the users. These records show the behavior of the users. Using auditing log, malicious users are easily traceable. Policy management is implemented properly to make the system efficient. . It supports passive work flow using fix roles and active work flow using varying attributes. Using this method, access control can be applied on huge number of users. I.e. it supports scalability which is very important feature of cloud environment.

It is clearly indicated from Table 1 & Fig. 4 that PAR-AC has added some new features too. In this method policy based authentication has been added to encase the valid /unique user on the website. It also decreases the traffic of non- valid user and saves website from various attacks such as D-Dos attack. Further, it reduces the complexity of the system and increases the efficiency, as different type of users has to pass through different type of authentication. Many researchers have proposed different access control model as discussed in introduction but they did not link it with authentication. Although AC3 method has proposed an idea of authentication function, it is not implemented yet and also it does not support any concept of policy based authentication. Proposed model is the only model which checks the type of users and then decides the type of authentication.

PAR-AC is user centric instead of application centric. Once the user is authenticated using policy based authentication; he can access all the modules or application for which he is authorized.

Moreover in the database data is encrypted and also attached with security tags and sensitivity levels, which make the system more secure.

PAR-AC also has the feature of role delegation, which helps to manage the system in absence of any higher position person. Such as in hospital management system, if the HOD of a particular department wants to go on leave, he can delegate his role to one of the senior doctors of the same department for the fixed time period. This feature has not been implemented by HA-RAC.

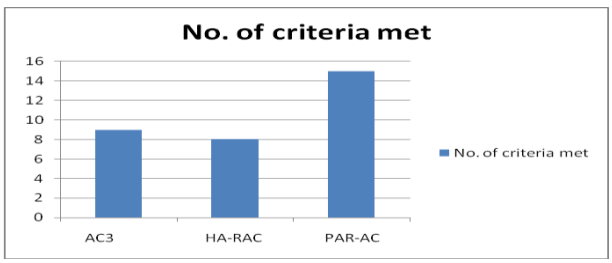


Fig. 4: Methods vs Number of Criteria Met.

Table 1: Comparison of PAR-AC with Existing Methods

Sr. No.	Comparison Criteria	AC3 [27]	HA-RAC [7]	PA-RAC
1	Least privilege assignment	1	1	1
2	Division of Duties	1	1	1
3	Configuration flexibility	1	1	1
4	Policy Management	1	1	1

5	Active and passive work flow support	1	1	1
6	Dealing with Hetero-geneity	1	1	1
7	Scalability	1	1	1
8	Implementation of the concept of fixed and variable attrib-utes for deciding roles	0	1	1
9	Maintaining of audit-ing log	1	0	1
10	Delegation of duties	1	0	1
11	Integration with Authentication Func-tions	Proposed but not implemented	0	1
12	Policy based Authen-tication	0	0	1
13	Backup of Data	0	0	1
14	Data encryption	Not dis-cussed	Not dis-cussed	1
15	User centric single sign on feature	Not dis-cussed	Not dis-cussed	1

PAR-AC also removes the loopholes of previous methods[7] [27],of not storing backups. Since heavy data can be accessed by the user and this makes the system slower; if the back up for all data is stored. But PAR-AC solves both the problems: it stores the backup but system will not be slow or heavy because back ups are stored for only a particular session. After ending of a particular session, all fetched data will be erased. Session helps to store the data till the session got abandon. A query with SQL server and encryption/decryption algorithm takes time to load data on every page. To increase the speed and performance of the pages, data are saved with different ways using sessions and view data. Session will provide only authenticated data and neglect the queries for the same page load which help us to save time. Results of efficiency increment, noted from hospital management module are shown below in Table 2 and Fig. 5.

Table 2: Improvement of Efficiency of the System with Storing Backups

Page Name	PAR-AC With Backup(in ms)	[7], [27] Without Backup(in ms)	Efficiency increment
AboutUS	0.00458	0.0596	92.32
Check Doctor Details	0.00468	0.0487	90.39
Check Previous Appointments	0.00582	0.0657	91.14
Approve Ap-pointments	0.00257	0.0652	96.06
Create Appoint-ment	0.00685	0.0527	87.00

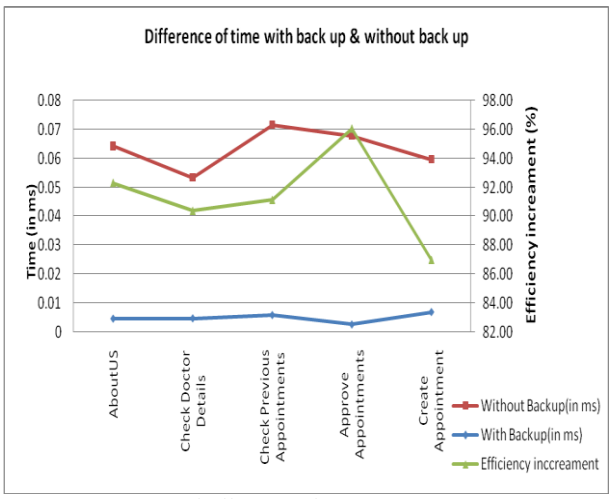


Fig. 5: Improvement of Efficiency of the System with Storing Backups.

5. Conclusion

In this paper we have proposed a new access control method which makes the system more secure with increased the efficiency of the system. We are confident that proposed method fulfils the major requirements of the cloud environment. It uses least privilege principle, maintains auditing logs and proper policy management policies which results in the increase of robustness of the system. The combination of fixed and varying attributes, sensitivity levels and security tags, play very important role in restricting the unlimited access to the data. PAR-AC includes important features of previously proposed methods and incorporates some new features which were missing in previous methods. It is proven to be better and an efficient method for cloud environment.

References

[1] Z. Shen and Q. Tong, "The security of cloud computing system enabled by trusted computing technology," *2010 2nd Int. Conf. Signal Process. Syst.*, pp. V2-11-V2-15, 2010. https://doi.org/10.1109/ICSPS.2010.5555234.

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in Cloud Computing," *2009 17th Int. Work. Qual. Serv.*, Jul. 2009, pp. 1–9.

[3] L. Youseff, M. Butrico, and D. Da Silva, "Toward a Unified Ontology."

[4] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, 2008, p. 50. https://doi.org/10.1145/1496091.1496100.

[5] S. Ullah and Z. Xuefeng, "Cloud Computing Research Challenges," in *proceedings of 5th IEEE International Conference on Biomedical Engineering and Informatics*, 2012, pp. 1397–1401.

[6] S. Ullah, Z. Xuefeng, and Z. Feng, "TCLOUD : A Multi – Factor Access Control Framework for Cloud Computing," vol. 7, no. 2, 2013, pp. 15–26.

[7] T. Cai, J. Zheng, and X. Du, "A hybrid attribute based RBAC model," *Int. J. Secur. its Appl.*, vol. 9, no. 7, 2015,pp. 317–328.

[8] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, and T. Freeman, "A flexible attribute based access control method for grid computing," *J. Grid Comput.*, vol. 7, no. 2, 2009, pp. 169–180. https://doi.org/10.1007/s10723-008-9112-1.

[9] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," *2010 Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9. https://doi.org/10.1109/INFCOM.2010.5462174.

[10] M. Sarfraz, A. A. Almutairi, and M. I. Sarfraz, "Access Control Architecture for Cloud Computing," no. October 2014, 2012.

[11] A. R. Khan, "Access Control in Cloud Computing Environment," vol. 7, no. 5, 2012, pp. 1–4.

[12] H. Andal and J. Narayanan, "Healthcare Systems," *1st IEEE Int. Work. Consum. eHealth Platforms, Serv. Appl.*, 2011, pp. 247–251.

[13] O. State, G. Polytechnic, and O. State, "A Modified Things Role Based Access Control Model for Securing Utilities in Cloud Computing," vol. 5, no. 2, 2015, pp. 21–25.

[14] Z. Tianyi, L. Weidong, and S. Jiaxing, "An Efficient Role Based Access Control System for Cloud Computing," *2011 IEEE 11th Int. Conf. Comput. Inf. Technol.*, 2011, pp. 97–102. https://doi.org/10.1109/CIT.2011.36.

[15] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy preserving access control with authentication for securing data in clouds," *Proc. - 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2012*, pp. 556–563. https://doi.org/10.1109/CCGrid.2012.92.

[16] A. Anderson, "Core and hierarchical role based access control (RBAC) profile of XACML v2. 0," *OASIS Stand.*, no. February, 2005, pp. 1–23.

[17] R. Oza, R. Patel, and A. Desai, "An Application of Hybrid ACL Approach in Cloud Environment," *Elsevier*, 2013.

[18] P. B. R. Kavali, "Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing," vol. 3, no. 2, 2013, pp. 743–754.

[19] D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role-based access control," *Computer (Long. Beach. Calif).*, vol. 43, no. 6, 2010, pp. 79–81. https://doi.org/10.1109/MC.2010.155.

[20] M. Li, S. Yu, K. Ren, and W. Lou, "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings," 2010, pp. 89–90.

[21] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data

Sharing with Attribute Revocation," *Proc. 5th ACM Symp. Information, Comput. Commun. Secur.*, 2010, pp. 261–270. https://doi.org/10.1145/1755688.1755720.

[22] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," *Proc. 17th ACM Conf. Comput. Commun. Secur. - CCS '10*, 2010, p. 735. https://doi.org/10.1145/1866307.1866414.

[23] F. Zhao, T. Nishide, and K. Sakurai, "Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6672 LNCS, 2011, pp. 83–97. https://doi.org/10.1007/978-3-642-21031-0_7.

[24] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," *Proc. 10th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2011, 8th IEEE Int. Conf. Embed. Softw. Syst. ICESS 2011, 6th Int. Conf. FCST 2011*, 2011, pp. 91–98. https://doi.org/10.1109/TrustCom.2011.15.

[25] W. Wang, J. Han, M. Song, and X. Wang, "The Design of a Trust and Role Based Access Control Model in Cloud Computing," *Ieee*, 2011, pp. 330–334. https://doi.org/10.1109/ICPCA.2011.6106526.

[26] J. Chen and T. Zhang, "Research and Implementation of Role-Based Access Control Model Based on Partition Number," 2009, pp. 6–9.

[27] Y. A. Younis, K. Kifayat, and M. Merabti, "An access control model for cloud computing," *J. Inf. Secur. Appl.*, vol. 19, no. 1, Feb. 2014,pp. 45–60. https://doi.org/10.1016/j.jisa.2014.04.003.

[28] U. Mukhtar and A. Naushahi, "Profile-Based Access Control in Cloud Computing Environments with applications in Health Care Systems," no. February, 2016.

[29] A. Chen, H. Xing, K. She, and G. Duan, "A Dynamic Risk-Based Access Control Model for Cloud Computing," *2016 IEEE Int. Conf. Big Data Cloud Comput. (BDCloud), Soc. Comput. Netw. (SocialCom), Sustain. Comput. Commun.*, no. 2014, 2016, pp. 579–584. https://doi.org/10.1109/BDCloud-SocialCom-SustainCom.2016.90.