WAS Science Nature
WASSN

ISSN: 2766 - 7715
WAS PRESS

Article

Design Access Control Model for the Cloud Computing Environment

Bassam. W. Aboshosha[1,*], Rabie. A. Ramadan[3], Ayman El-Sayed[2], and Mohamed M. Dessouky[2]

[1] Department of Computer Engineering, Higher Institute of Engineering, Elshorouk Academy, Egypt
[2] Department of Computer Engineering, Menofia University, Egypt
[3] Computer Engineering Dept. Cairo University, Cairo, Egypt and University of Hail, Hail, KSA.
* Correspondence: bassam.ahmed32@gmail.com

Correspondence author: Bassam. W. Aboshosha1

Received: date; Accepted: date; Published: date

Abstract: Recently, cloud computing applications are growing very fast, with the increasing number of organizations that resorting to use or store resources in the Cloud, several challenges have been identified. Security is one of the most challenging aspects of Cloud Computing. Access control offers strong security for data and is considered as a major research area. A good access control model assures a secure cloud environment. This research aims to introduce a user profile based access control model in cloud computing environments and architecture.

Keywords: Access control model; cloud computing; tokenization; access list.

1. Introduction

Cloud computing is a promising computing paradigm that recently has taken extensive attention from both the academic world and commerce. By merging a set of current and new techniques from research areas such as Service-Oriented Architectures (SOA) and virtualization, cloud computing is regarded as such a computing paradigm in which resources in the computing infrastructure are provided as on-demand services over the Internet. Along with this new paradigm, several business models are established, which can be described by the terminology of "X as a service (XaaS)" [1] where X could be infrastructure, platform, software, data storage, etc. Successful examples are Amazon's EC2 and S3, Google App Engine, and Microsoft Azure which provide users with scalable resources in a pay-as-you-use manner at relatively low cost. For example, Amazon's S3 data storage service just pay $0.12 to $0.15 per Giga by the month. Comparing to building their infrastructures, customers (organizations/persons) can save their investments significantly by migrating businesses into the cloud. With the increasing development of cloud computing technologies, it is easy to predict that soon more and more businesses will be moved into the cloud.

On the other hand, cloud computing is also suffering from many challenges that, if not well fixed, may Threaten its fast growth. Information security, as it exists in many other applications, is one of these challenges that would take great concern from users when they store sensitive information on cloud servers. These concerns are formed from the fact that cloud servers are usually operated in commercial environments and managed by business providers which are very likely to be outside of the trusted domain of the users.

Data secrecy against cloud servers is hence frequently desired when users outsource data for storage in the cloud. In some practical application systems, data confidentiality is not only a security/privacy issue but also of juristic concerns. For example, in healthcare application scenarios use and disclosure of protected health information (PHI) should meet the requirements of the Health

Insurance Portability and Accountability Act (HIPAA) and keeping user data confidential against the storage servers is not just an option, but a necessity.

Furthermore, we observe that there are also cases in which cloud users themselves are content providers. They publish data on cloud servers for sharing and need fine-grained data access control in terms of which user (data consumer) has the access privilege to which types of data. In the healthcare case, for example, a hospital or clinic would be the data owner who stores millions of healthcare records in the cloud. It would allow data consumers such as doctors, nurses, patients, researchers, etc., to access several types of healthcare records under policies admitted by HIPAA. To carry out these access policies, the data owners, on one hand, would like to take advantage of the abundant resources that the cloud provides for efficiency and economy; on the other hand, they may want to keep the data contents confidential against cloud servers.

As a significant research area for system protection, data access control has been growing in the past thirty years and several techniques have been developed to effectively implement fine-grained access control, which allows flexibility in specifying differential access rights of individual users. Traditional access control architectures usually assume the data owner and the servers storing the data are in the same trusted domain, where the servers are fully entrusted as an omniscient reference monitor responsible for defining and enforcing access control policies. This assumption however no longer holds in cloud computing since the data owner and cloud servers are very likely to be in two different domains. On one hand, cloud servers are not allowed to access the outsourced data content for data confidentiality; on the other hand, the data resources are not physically under the full control of the owner. To help the data owner enjoy fine-grained access control of data stored on untrusted cloud servers, a feasible solution would be encrypting data through certain cryptographic primitive(s) and disclosing decryption keys only to authorized users. Unauthorized users, including cloud servers, are not able to decrypt since they do not have the data decryption keys. This general method has been widely adopted by existing works [2][3] which aim at securing data storage on untrusted servers. One critical issue with this branch of approaches is how to achieve the desired security goals without introducing a high complexity on key management and data encryption. These existing works, resolve this issue either by introducing a per-file access control list (ACL) for fine-grained access control or by categorizing files into several filegroups for efficiency. As the system scales, however, the complexity of the ACL-based scheme would be proportional to the number of users in the system. The filegroup-based scheme, on the other hand, is just able to provide coarse-grained data access control.

Our paper relates to the newer concept of "Security as a Service", where security services are provided to Cloud users [4][5]. More specifically, we develop the concept of Access Control as a Service". The main goal of access control is to regulate the different operations that may be performed by subjects on objects. Access control models provide a formal representation of access control policies and are the result of constant evolution in security requirements. Hence, over the last four decades, many approaches to access control have been developed, in this paper, we present a new approach, for building dedicated access control models. Hence, in our approach, following the high-level security policies of each specific user. Our framework for building access control models can be implemented as a Cloud service and Cloud providers will then apply different concrete access control models produced by each user to process its incoming access requests.

The rest of the paper is organized as follows. Section 2 various approaches to access control have been described, section 3 describes NIST cloud computing architecture, section 4 presents the problem formulation, section 5 introduced the proposed access control model, and concluding remarks are made in Section 6.

2. Access Control Models

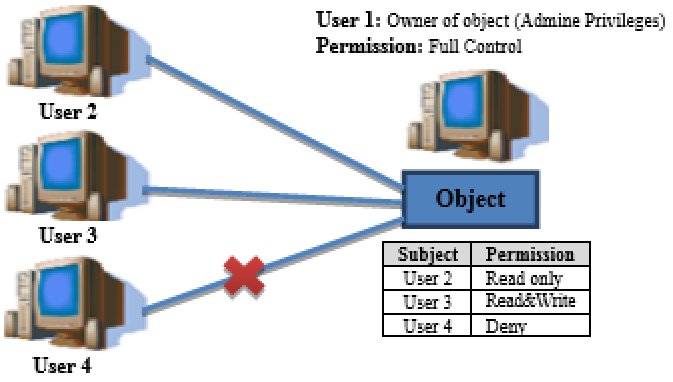
Access control is a fundamental aspect of information security that is directly tied to the primary characteristics such as confidentiality, authentication, integrity, and availability. Cloud computing service providers should provide the following basic functionalities from the perspective of access control:

1. Control access to the service features of the cloud-based on the specified policies and the level of service purchased by the customer.
2. Control access to a consumer's data from other consumers in multi-tenant environments.
3. Control access to both regular user functions and privileged administrative functions.
4. Maintain accurate access control policy and up to date user profile information.

Access control models can be traditionally categorized into three types: Discretionary access control (DAC), Mandatory access control (MAC), and Role based access control (RBAC).

These models are explained with the help of simplified model diagrams, also listed the advantages and disadvantages of the access control models, which could help to select an access control model according to the need and type of the access required to set up the cloud environment.

1) **Discretionary access control (DAC):** This is the conventional access control in which client has the complete control over the project. DAC is based on offering access to the client on the premise of client character and approval which is characterized for open policies [6]. DAC policy depends on the client's identity and authorization that indicates for every client's access strategy and object that is requested by client. DAC has access attributes and access rules, the access attributes permits the system to define various authorization levels, and the access rules prevent the unauthorized clients to access any information [7][8].



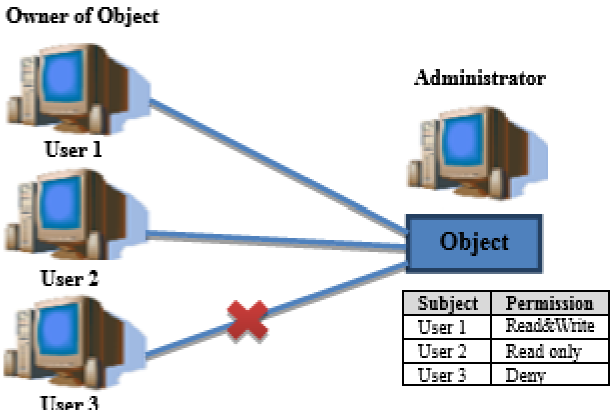
Subject	Permission
User 2	Read only
User 3	Read&Write
User 4	Deny

Figure 1. Model diagram of DAC

Advantage: Adaptability in utilization by keeping up the authorization database which comprises of authorized clients.

Disadvantage: This model provides neither high-level security nor risk awareness. DAC is highly prone to attacks like the Trojan horse virus attack. The worst part is that it is not compatible with cloud computing implementation.

2) **Mandatory access control (MAC):** Mandatory access control upholds the control based on directions by the administrator. The most common form of mandatory access control is the multilevel security access control, based on subjects and objects in the system. Objects act as passive entities and store data while Subjects acts as active entities and sends an access request to the objects [6]. A central administrator assigns a security level on data based on its significance and sensitivity. It additionally assigns a security clearance to users. Only the users with a security clearance level more prominent than the security level of the data can have access to the data [7] [8].



Subject	Permission
User 1	Read&Write
User 2	Read only
User 3	Deny

Figure 2. Model diagram of MAC

Advantages: As only the system administrator can access or change the control, MAC provides strong security, and also it reduces security errors.

Disadvantages: It doesn't promise complete protection of data. The system administrator is vulnerable to attacks since it is the one who knows about the security levels. Additionally, this model is costly to implement [9].

3) **Role-based access control (RBAC):** Role-based access control upholds the control based on the role assigned to the user. The user's role determines his security clearance [6] [7]. RBAC model can be utilized to execute a few essential security standards, for example, least privilege, separation of duties, and data abstraction. RBAC has various administrative policies such as centralized, hierarchical, co-operative, ownership, and decentralized [8][10][11].

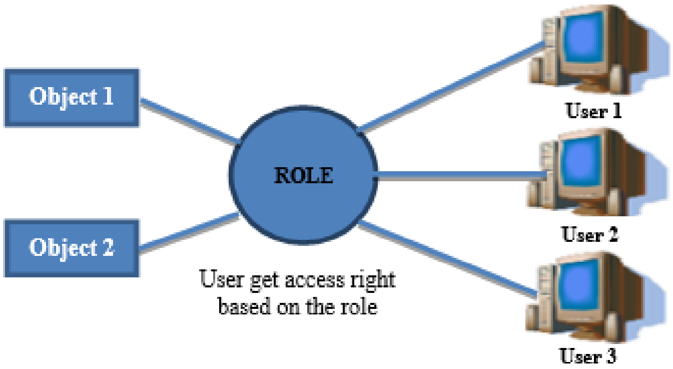


Figure 3. Model diagram of RBAC

Advantages: As roles are assigned on the minimum privilege access, it decreases the level of damage. Separation of roles limits the chance of misuse of data.

Disadvantages: Users with multiple roles could use one role to grant access to an object which is restricted for his other role. This is a violation of security policy.

3. NIST Cloud Computing Architecture

Based on the definition introduced by Peter et al. [14] which is widely accepted among researchers from the academic world and commerce, researchers at NIST formulated the conceptual architecture for the cloud computing model. The architecture presents a high-level conceptual view of the cloud computing model and identifies the major actors along with their activities and functions.

It serves as a reference model for researchers to study the cloud computing model and its various components to design and simulate the cloud-based services. The reference architecture is presented in Figure 4. It incorporates five important entities that play an important role in the development of cloud computing models.

Cloud Consumer: The cloud consumer is a person or organization that maintains a business relationship with a cloud provider, and uses its services offered on a cloud. Examples include hospitals, schools, etc.

Cloud Provider: The cloud provider is a person, organization, or entity responsible for making a cloud service available to interested parties. Examples include Google, Microsoft, Amazon Web service, etc.

Cloud Carrier: The cloud carrier is a party that can conduct an independent assessment of cloud services, information system operations, performance, and security of the cloud implementation.

Cloud Auditor: The cloud auditor is an entity that manages the use, performance, and delivery of cloud services, and negotiates relationships between cloud providers and cloud consumers.

Cloud Broker: The cloud broker is an intermediary that provides connectivity and transport of the cloud services from cloud providers to cloud consumers.

Numerous researchers have adopted this architecture to design and simulate cloud computing models. Major research has been invested to improve the components presented at the cloud provider end. Our effort is no exception, as we present an access control-based security solution to improve the security and privacy of a cloud computing environment.

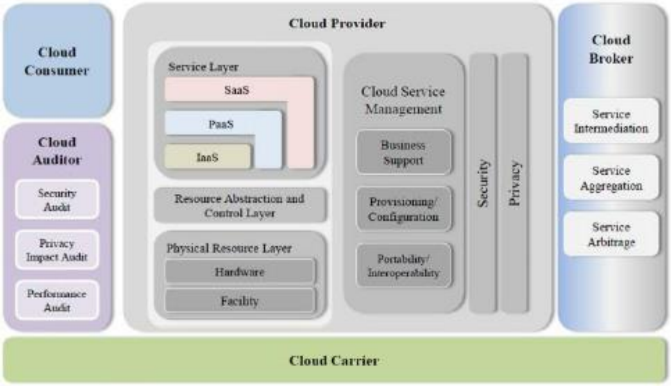


Figure 4. NIST Cloud Computing Reference Architecture

4. Problem Formulation

Buyya et al. [12] who define a Cloud as a \type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers".

This ambitious approach, considering at the same time several forms of deployment and a wide spectrum of features, emphasizes the use of information technology decorrelating the applications and other infrastructure resources underlying its mechanism of distribution.

When data and systems are hosted in shared hosting environments, access control to data, data privacy, and data separation become crucial [13]. Hence, access control can play an important part in data protection needs in Cloud environments.

Cloud users can manage (create, write, edit, etc) their data on various cloud platforms like banking apps, Office 360, Sky Drive, Dropbox, etc. These applications use very large amounts of data, which is saved with complex storage architecture. Various users access different patterns of information on these cloud platforms. The access control authentication can be used to divide the user data access control up to various stages based on multi-level authentication schemes. This will ensure the security of the data storage on the cloud platforms. In order to access these cloud platforms from the touch-based devices, the users face difficultly in providing the different levels of text-based passwords. We are trying to improve the user experience on the touch-based devices using a multi-tier access control authentication using the graphical techniques of different types.

One critical issue with this branch of approaches is how to achieve the desired security goals without introducing a high complexity on key management and data encryption. These existing works, resolve this issue either by introducing a per-file access control list (ACL) for fine-grained access control or by categorizing files into several filegroups for efficiency. As the system scales, however, the complexity of the ACL-based scheme would be proportional to the number of users in the system.

Umair Mukhtar [15] introduces the Profile-Based Access Control system. He supposed that Once the user profile is validated the access token is granted to the user for all the services that are available to the user of that particular profile. It represents a security compromise to all services, in this case, once intruder gain access he/she can alter or effect on the data at least he/she gain access to all available data in each service.

We recommend in our approach that, it should be added a security service stage to perform authorizations for each service. Each service should have its verification code, user has to be know this code to gain access to certain service. Also, it improves the overall security of the cloud as only service which are available for that particular profile and exceed the final security stage is exposed to the user. The final security stage is added after the system process diagram that presented in [15].

5. Proposed Access Control Model

The presentation of our solution is based on the layered NIST architecture [14] as described earlier. The solution is presented at the security layer which is the gateway to access the cloud services. The major contribution of our solution is an improved access controller. There are three main components in the proposed access control system.

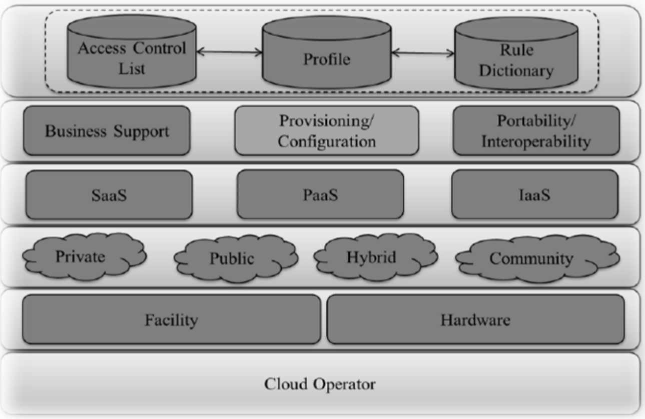


Figure 5. Proposed Cloud Computing Security Architecture

5.1. Profiles

Profile or user personalization is not a new concept in the computing world. Nowadays personalization has widely been used to personalize content and services according to the user's interests and preferences. The personalization attributes are used to present users with content or services that match their profiles. In the cloud computing environment, it is highly desirable to have automatic mechanism in place that can verify the security and access policies of services and resources for the cloud users.

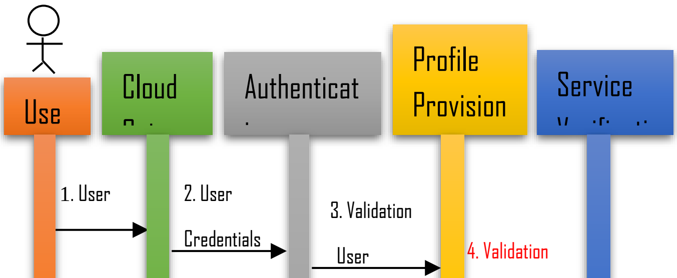
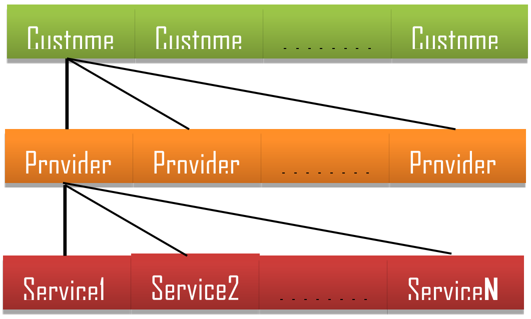


Figure 6. Proposed System Process Diagram

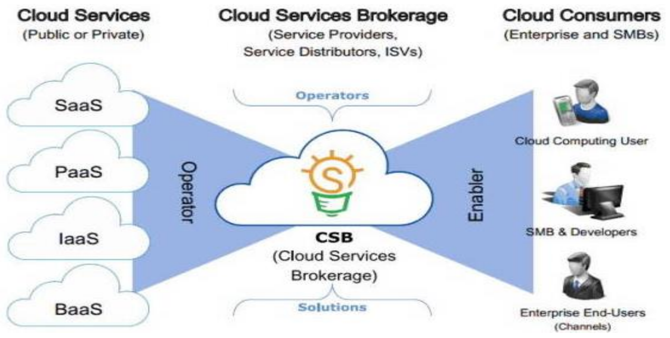
In our solution, we used a profile-based access control system that can be used to define security and access policies for the services and resources offered on the cloud. The profile in our case is an entity that has some pre-defined privileges to access cloud services. When a user accesses the cloud and provides the credentials, the authentication system in place validates the user credentials and verifies the user profile. Once the user profile is validated user allowed to access the last verification stage to enable the user to access a certain service. this security service stage is used to perform authorizations for each service. Each service should have its verification code, the user has to know this code to gain access to certain services. Once the user inserts a valid service code, a service access token is generated and delivered to the user. the access token is granted to the user for the requested service in which code has been inserted and matched. User is allowed to access that service and deal with the service according to its privileges policies and specific rules.

Also, it improves the overall security of the cloud as the only service which is available for that particular profile and exceeds the final security stage is exposed to the user. The system process diagram is presented in Figure 6.

As a first step in the process, the user sends his/her credentials to the cloud gateway, where the authentication service will validate the user. Once the user is validated, the profile provisioning service validates the profile, service verified, and generates a user's service access token. The access token expires once the user signs out from the cloud or when the access session expires. Users, providers, and their services hierarchy is shown in figure 7.



(a)



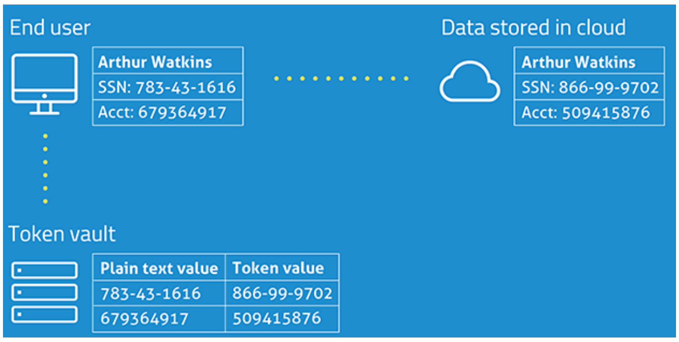
(b)

Figure 7. Hierarchy of users, providers and their services

5.2. Tokenization

Tokenization and encryption are often mentioned together as a means to secure information when it's being transmitted on the Internet or stored at rest. In addition to helping to meet your organization's data security policies, they can both help satisfy regulatory requirements such as those under PCI DSS, HIPAA-HITECH, GLBA, ITAR, and the EU GDPR. While tokenization and encryption are both effective data obfuscation technologies, they are not the same thing, and they are not interchangeable. Each technology has its strengths and weaknesses and based on these, one or the other should be the preferred method to secure data under different circumstances. In some cases, such as with electronic payment data, both encryption and tokenization are used to secure the end-to-end process. In our research, we interest to understand the tokenization process.

Tokenization is the process of turning a meaningful piece of data, such as an account number, into a random string of characters called a token that has no meaningful value if breached. Tokens serve as a reference to the original data, but cannot be used to guess those values. That's because, unlike encryption, tokenization does not use a mathematical process to transform sensitive information into the token. There is no key or algorithm, that can be used to derive the original data for a token. Instead, tokenization uses a database, called a token vault, which stores the relationship between the sensitive value and the token. The real data in the vault is then secured, often via encryption. The figure8 shows the Tokenization process.



Arthur Watkins
SSN: 783-43-1616
Acct: 679364917

Arthur Watkins
SSN: 866-99-9702
Acct: 509415876

Plain text value	Token value
783-43-1616	866-99-9702
679364917	509415876

Figure 8. The Tokenization process

The token value can be used in various applications as a substitute for the real data. If the real data needs to be retrieved for example, in the case of processing a recurring credit card payment – the token is submitted to the vault and the index is used to fetch the real value for use in the authorization process. To the end-user, this operation is performed seamlessly by the browser or application nearly instantaneously. They're likely not even aware that the data is stored in the cloud in a different format.

The advantage of tokens is that there is no mathematical relationship to the real data they represent. If they are breached, they have no meaning. No key can reverse them back to real data values. Figure 9 illustrates a service (application) access control scenario based on the token.

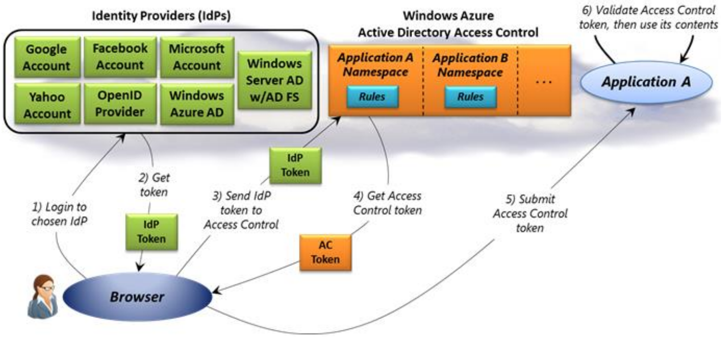

Identity Providers (IdPs)
Google Account
Facebook Account
Microsoft Account
Windows Server AD w/AD FS
Yahoo Account
OpenID Provider
Windows Azure AD
Windows Azure Active Directory Access Control
Application A Namespace
Application B Namespace
Rules
Rules
...
6) Validate Access Control token, then use its contents
Application A
1) Login to chosen IdP
2) Get token
IdP Token
3) Send IdP token to Access Control
IdP Token
4) Get Access Control token
AC Token
5) Submit Access Control token
Browser

Figure 9. illustrates an application access control based on tokenization

5.3. Rule Dictionary

The rule dictionary is proposed to define the security policy of the system. It defines the access privileges for all the resources offered by the cloud. These privileges are defined in the rule dictionary for each service the profile can access. Traditionally rules were separately defined for every CURD (Create, Update, Retrieve, and Delete) operation, thus increasing the number of entries in the rule book and so its management cost. In our solution, a rule identifier is defined and stored against each service and profile.

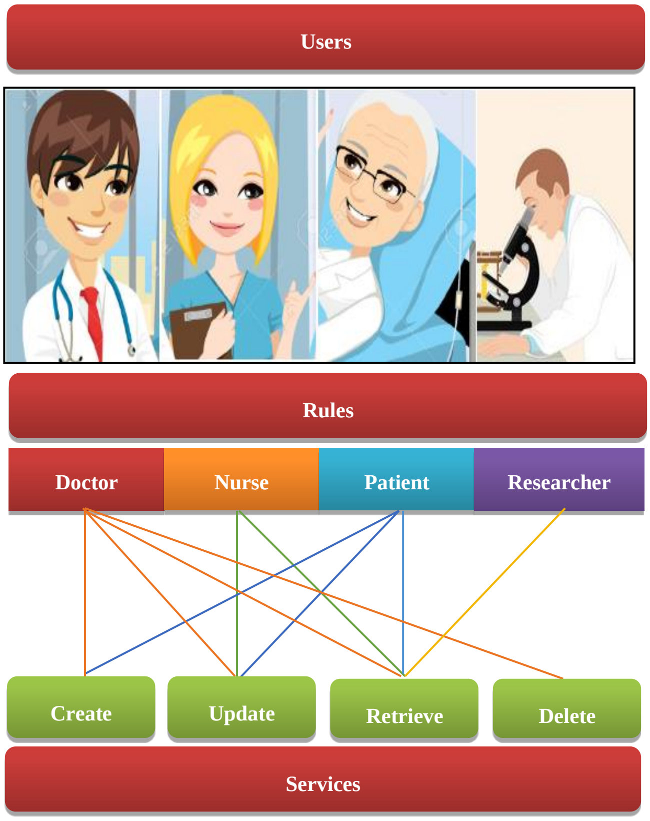


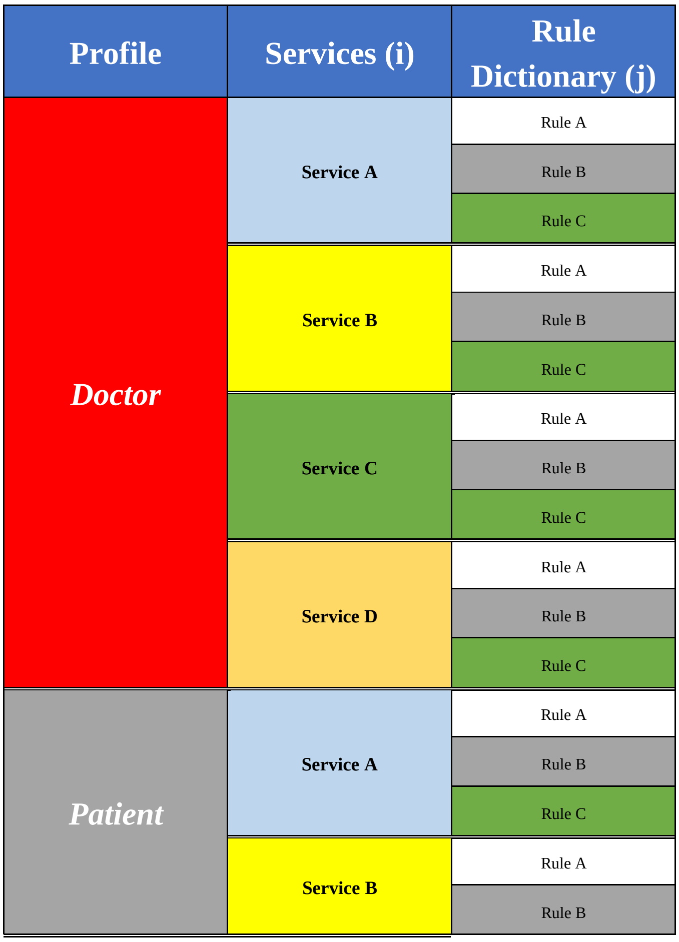
Figure 10. Illustration of Role-based Access Control System for dictionary rules A.

The access policy (i.e. CURD operations) of these identifiers is stored in the rule dictionary. The identifier reads the rule dictionary and executes the security policy defined for each service and profile. The benefit of this approach is that it allows system administrators to update the rule dictionary for each rule identifier without affecting the access control list. This also reduces the management and implementation cost of ACL. In addition to CURD operations, the rule dictionary can also define access rules for different deployment models. This helps improve service interoperability and access rights across cloud networks. For simplicity, we have only incorporated two deployment models (public, private). However, this can all be extended based on cloud operator requirements. Figure10 Illustrates the Role-based Access Control System.

5.4. Access Control List

The services that are accessible to a particular profile and the access rules associated with it are defined in the access control list with the following data structure: Profile (i, j) where i represent a service, and j represents the rule identifier. An example of such an ACL entry is Patient (PHR, Rule Rth), A more complete example of the access control list is shown in Figure 11a.

The description of each service and resource are shown in Figure 11b. All new services and resources offered in the cloud will be added to this list. Figures 11c, 11d, and 11e show the entries for access rights and privilege entries defined for each profile along with the respective rules for Services A, B and C. Finally, the rules and access privileges for each service are defined in the rule dictionary. Traditionally, the access list is maintained as one big list, which increases the complexity of maintaining it. In the proposed solution, we have decomposed the list in different parts. The benefit of such a decomposition is a reduction in management and administration costs. This decomposed version is also easier to implement in a cloud computing environment where rules can be offered as a service.



Profile	Services (i)	Rule Dictionary (j)
Doctor	Service A	Rule A
		Rule B
		Rule C
	Service B	Rule A
		Rule B
		Rule C
	Service C	Rule A
		Rule B
		Rule C
	Service D	Rule A
		Rule B
		Rule C
Patient	Service A	Rule A
		Rule B
		Rule C
	Service B	Rule A
		Rule B

		Rule C
	Service C	Rule A
		Rule B
		Rule C
	Service D	Rule A
		Rule B
		Rule C
Nurse	**Service A**	Rule A
		Rule B
		Rule C
	Service B	Rule A
		Rule B
		Rule C
	Service C	Rule A
		Rule B
		Rule C
	Service D	Rule A
		Rule B
		Rule C
Researcher	**Service A**	Rule A
		Rule B
		Rule C
	Service B	Rule A
		Rule B
		Rule C
	Service C	Rule A
		Rule B
		Rule C
	Service D	Rule A
		Rule B
		Rule C

(a) Profile-based cloud Access Control List for CURD Services.

Service ID	**Services Resources Description**
Service A	Create a patient account (C)
Service B	Update patient information (U)
Service C	Retrieve patient information (R)
Service D	Delete patient account (D)

(b) Service and resource list

Profile	**Privileges**				**Deployment Model**	
	C	*U*	*R*	*D*	*Public*	*Private*
Doctor	*Y*	*Y*	*Y*	*Y*	*Allow*	*Allow*
Patient	*Y*	*Y*	*Y*	*N*	*Deny*	*Allow*
Nurse	*N*	*Y*	*Y*	*N*	*Allow*	*Allow*
Researcher	*N*	*N*	*Y*	*N*	*Allow*	*Deny*

(c) Rule Ath Dictionary

Profile	**Privileges**				**Deployment Model**	
	C	*U*	*R*	*D*	*Public*	*Private*
Doctor	*Y*	*Y*	*Y*	*Y*	*Allow*	*Allow*
Patient	*N*	*N*	*Y*	*N*	*Deny*	*Allow*
Nurse	*Y*	*Y*	*Y*	*N*	*Allow*	*Allow*
Researcher	*N*	*N*	*Y*	*N*	*Allow*	*Allow*

(d) Rule Bth Dictionary

Profile	**Privileges**				**Deployment Model**	
	C	*U*	*R*	*D*	*Public*	*Private*
Doctor	*Y*	*Y*	*Y*	*Y*	*Allow*	*Allow*
Patient	*Y*	*N*	*Y*	*N*	*Deny*	*Allow*
Nurse	*Y*	*N*	*Y*	*N*	*Allow*	*Deny*
Researcher	*N*	*N*	*Y*	*N*	*Deny*	*Deny*

(e) Rule Cth Dictionary

Figure 11. Presenting the profile-based Access Control List

6. Conclusion

This paper presented a new profile based access control system. The solution is proposed at the security layer of the cloud computing architecture. The management of ACL requires less administrative effort as compared to traditional solutions. The profile management and rule assignment are less complex. Besides, our solution offers reduced data access time and cost. Also, considering the hierarchical organizational structure of the health care system, the profile based access control approach best fit.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., ... & Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing (Vol. 17). Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley.
2. Kallahalla, M., Riedel, E., Swaminathan, R., Wang, Q., & Fu, K. (2003, March). Plutus: Scalable Secure File Sharing on Untrusted Storage. In Fast (Vol. 3, pp. 29-42).
3. Di Vimercati, S. D. C., Foresti, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2007, September). Over-encryption: Management of access control evolution on outsourced data. In Proceedings of the 33rd international conference on Very large data bases (pp. 123-134).
4. Yu, T., & Winslett, M. (2003, May). A unified scheme for resource protection in automated trust negotiation. In 2003 Symposium on Security and Privacy, 2003. (pp. 110-122). IEEE.
5. Blaze, M., Bleumer, G., & Strauss, M. (1998, May). Divertible protocols and atomic proxy cryptography. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 127-144). Springer, Berlin, Heidelberg.
6. Sohal, I. S., & Kaur, (2016). A. Review on advanced access control models in cloud computing.
7. Punithasurya, K., & Priya, S. J. (2012). Analysis of different access control mechanism in cloud. International Journal of Applied Information Systems, 4(2), 34-39.
8. Samarati, P., & de Vimercati, S. C. (2000, September). Access control: Policies, models, and mechanisms. In International School on Foundations of Security Analysis and Design (pp. 137-196). Springer, Berlin, Heidelberg.
9. Ajgaonkar, S., Indalkar, H., & Jeswani, J. (2015). Activity based access control model for cloud computing. International Journal of Current Engineering and Technology, 5(2), 708-713.
10. Ramadan, R. A. (2021). Detecting adversarial attacks on audio-visual speech recognition using deep learning method. International Journal of Speech Technology, 1-7.
11. Kamal, S., Ramadan, R. A., & Fawzy, E. R. (2015). Smart outlier detection of wireless sensor network. Facta Universitatis, Series: Electronics and Energetics, 29(3), 383-393.
12. Borkin, S. (2003). The HIPAA final security standards and ISO/IEC 17799. Collect. Information Security Reading Room, 25.
13. Fouad, M. M. M., El-Bendary, N., Ramadan, R. A., & Hassanien, A. E. (2013). Wireless sensor networks: a medical perspective. Wireless Sensor Networks: From Theory to Applications.
14. Peter, M. & Tim, G. (10 July 2009). The NIST definition of Cloud Computing, Version 15. Information Technology Laboratory. Online: http://www.hexistor.com/blog/bid/36511/The-NIST-Definition-of-Cloud-Computing
15. Ramadan, R. A. (2009, March). Agent based multipath routing in wireless sensor networks. In 2009 IEEE Symposium on Intelligent Agents (pp. 63-69). IEEE.