JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 07 Issue 01
2024

Research Article
https://doi.org/10.56979/701/2024

A Hybrid Effective Access Control in Cloud Computing

Mumtaz Hussain[1], Muhammad Hasnain[2], Asif Khanzada[3], Mohsin Ali Arif[4], Youssef AboShaban[5], Muhammad Asgher Nadeem[6], and Syed Muhammad Mohsin[7*]

[1]Department of Computer Science & IT, Institute of Southern Punjab, Multan, Pakistan.
[2]Department of Computer Science & IT, Ghazi University, Dera Ghazi Khan, Pakistan.
[3]Cloud Modernization Thomson Reuters – CTO & CIO ORG, Canada.
[4]JV with Saudi Aramco, Air Products & ACWA Power JIGPC, Saudi Arabia.
[5]Senior Technical Support Specialist, Ajman University, United Arab Emirates.
[6]Department of Computer Science & IT, Thal University, Bhakkar, Pakistan.
[7]Department of Computer Science, COMSATS University Islamabad, 45550, Pakistan.
*Corresponding Author: Syed Muhammad Mohsin. Email: syedmmohsin9@yahoo.com

Received: March 19, 2024 **Accepted:** May 21, 2024 **Published:** June 01, 2024

Abstract: Data security, access control, and integrity are considered one of the main obstacles to the further development of cloud storage services. This is because accessing and sharing data across multiple applications is a fundamental requirement for cloud storage. This sharing of information across the cloud also raises issues such as privacy, confidentiality, and unauthorized access. Building a secure storage system requires powerful access control and revocation management. Previously proposed attribute-based access control mechanisms have struggled with issues such as the complexity of privilege management, revocation, and credential verification. This study provides an efficient, simple, and flexible access control mechanism for cloud storage systems that fine-grains the attribute-based access control and role-based access mechanism.

Keywords: Cloud Access Control; Attribute-Based Access Control; Cloud Security.

1. Introduction

An access control model shows how information can be protected against [1] unauthorized use and threats to confidentiality (privacy) and unauthorized and incorrect modification and destruction (honesty). Customers (subjects) can be assured of the level of security by ensuring access decisions are guided by an access control strategy. Such an access control model consists of two critical components. [1] A mechanism for formulating the use case is necessary, as it is a component for determining the outcome of the request. The decision is made at the authority level. [2] Allow the grant request after it is received. The request may be compared to a previously established policy. The request is then evaluated to determine whether to accept it or reject it.

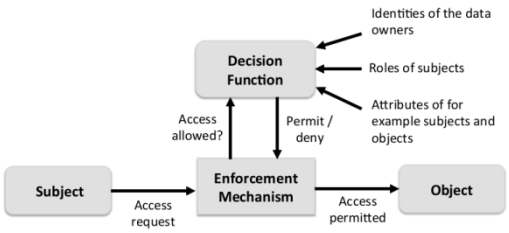

Figure 1. The Generic Access Model depicts the requirements system and decision-making capability

A cloud storage system consists of a collection of storage servers that provide long-term storage over the Internet. Information is stored and shared through a third-party system, which can raise questions about data security and the sharing of personal information. The various operations performed in the

existing approaches to access management with encrypted data may adversely affect the quality of the storage system. Generally, distributed storage system are built with a central authority that supports functions over the encrypted data.

Cloud-based delivery models enable significant cost savings. Cloud computing is expected to benefit large information technology companies by serving as a platform for third-party storage and service providers. IaaS, PaaS, SaaS, and S3 are the four main categories of these services, each of which has three subcategories: Infrastructure, Platform, and Storage (StaaS).

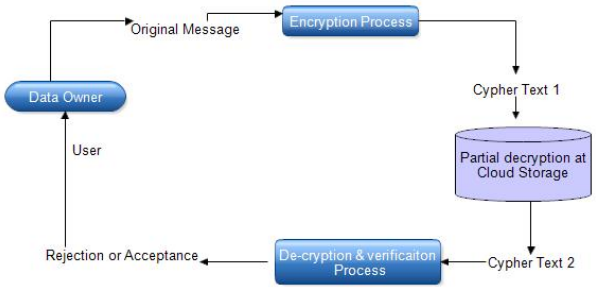

Figure 2. The encryption and decryption techniques used in cloud storage

Data access management is a cost-effective way to ensure cloud data security. Therefore, traditional server-based data access management solutions are no longer applicable to cloud storage systems due to data offloading. At times, archaic techniques are used to control access to information on untrusted servers. A potential solution is employing data encryption. Thus, only people with authentic keys are able to decrypt encrypted data. If the system has a large number of users, managing the keys for the different systems is quite tedious. In order to grant access to new users, the data owners must be active and online all the time. In addition, the cipher text must be written for a large number of users, with unique keys for each piece of information. This has a significant impact on the storage capacity of the server [3].

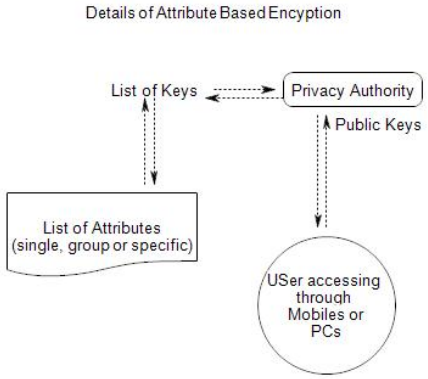

Figure 3. A Brief Overview of Attribute-Based Encryption

There are a number of challenges that users of cloud storage, both businesses and individuals, may face when using the service. The integrity and security of data storage is one of the fundamental challenges. Both users and cloud service providers need to manage the access control to address the full range of storage needs. While cloud storage security is critical in the context of cloud computing, this solution should also be simple and flexible enough to accommodate a diverse user base. Although several researchers have already conducted research in this area and proposed various approaches to access control, there are still difficulties to overcome. Although this is a one-time project, the end result will be a user-friendly attribute-based access mechanism.

2. Review of Literature

Yuan & Tong's 's (2005) study described the challenges of online access and focused on the subjects, associated items, and environment. It also offered a model for policy architecture and a mechanism for integrating permissions. The model is divided into two components: a structural model and a policy model. The policy model defines how policies are applied to and implemented in a web system. An example policy model is used to illustrate the policies that govern a person's access to a resource.

$$Rule\colon\ can\text{-}access(s,r,e) \leftarrow f(ATTR(s), ATTR(r), ATTR(e)).$$

Before providing access to the properties s, r, and e, the function checks whether the evaluation results are accurate. Yuan and Tong did not elaborate on operations on objects. In addition, they did not provide any context for policy implementation or evaluation.

A number of ABAC models for a variety of applications and domains are then presented. Alshehri et al. have extended the core attribute model for SOA access control (2013).

When a single repository is accessible to many programs, [4] provide a unified approval mechanism for different policy requirements.

By encrypting documents with symmetric encryption and transmitting keys with asymmetric encryption, [5] has developed a lightweight encryption algorithm.

One of its advantages is the requirement of fine-grained data access, which is a critical aspect of management. It is also not adaptable or versatile, as encryption and decryption is required when a client leaves the group, which prevents it from accessing the information.

According to [6], users spend a lot of time browsing multimedia files on the Internet. The study discusses cloud-based mobile multimedia recommendation systems that minimize the network overhead and speed up the suggestion process. On the other hand, the cache has limited capacity, which makes accessing files stored on the cloud server as challenging the major problem in the proposed method is data integrity [7]. In this model, the data owner verifies the cloud data. Therefore, the proposed solution provides significant security and data integrity to the cloud-based resources of the data owners.

Cryptographic methods are becoming more versatile with the inclusion of additional mathematical tools, often requiring numerous keys for a single application. [7] To enable the mapping of secret keys for different digit text classes to cloud storage, the authors of this study investigated the use of compression algorithms to "compress" secret keys in public-key cryptosystems. The maximum number of cipher text classes places a constraint on the authors' work. The amount of encrypted text stored in cloud storage is growing rapidly.

When it comes to avoiding the high costs associated with purchasing and maintaining expensive hardware, cloud storage services are proving to be a viable alternative. (2014) (Enrico Bocchi) The performance of cloud storage has been studied, in particular, through tests designed and conducted by this team. However, when sharing files with other services, it has limited ability to meet users' needs. This is because certain services have problems with utilisation, while others perform well or poorly.

Everaldo Aguiar (2014) studied the authentication, virtualization, availability, accountability, and privacy of remote storage and compute [8]. Thanks to these advanced protections, client data and computation can be kept private and secure. However, the fundamental problem with these methods is that they constantly lead to the discovery of new security vulnerabilities. Moreover, the confidentiality of the data limits the working conditions.

According to (Hsiao-Ying Lin, 2012), there are three main problems with current methods: First, the high volume of traffic and computation, then the management of cryptographic keys, and finally the storage servers that do not directly allow other operations such as forwarding messages between users. They have developed an encryption strategy that combines the use of an inefficient and costly threshold proxy encryption approach with an uncoordinated decentralized erasure key to building a secure distributed storage system. The downside of this scheme is that merging it with other schemes requires extensive analysis of the enterprise as a whole.

Multi-tenancy is often seen as an advantage by cloud service providers, but it also poses a security risk. In developing resource allocation strategies, the authors prioritized security without sacrificing performance, power consumption, or cost. Due to computational and time constraints, it was not possible for the authors to replicate the entire attack model.

Secure Data Sharing in Clouds (SeDaSC) is a technique [9] that provides data confidentiality and integrity, access management, data sharing (forwarding), and insider threat protection without requiring computationally intensive re-encryption. There is an issue with key management and key size evaluation.

The authors have developed data access control systems for multi-authority cloud storage systems (DAC-MACS) that are secure under lower security assumptions [10]. A mechanism for deciphering a new scheme for attributing multi-authority cipher text policies has been proposed (CP-ABE). However, the use of these methods increases the cost of computation and communication.

By using encryption and steganography on both the client and server sides, a very secure approach can be achieved [11]. The proposed approach has shortcomings and limitations that make it difficult to implement and use.

Storage, virtualization, and networking have been identified as the main sources of security threats [12]. Traditional security measures can be problematic in cloud environments because the cloud architecture is a complicated synthesis of multiple technologies, which is ultimately the most important limitation to the development and operation of any approach [13] has identified several critical future issues including lightweight data audits, dynamic data updates, data access management, and computational integrity.

Attribute revocation can be beneficial for multi-authority systems ABE (attribute-based encryption) [14]. However, it is costly because the data owner must transmit a new cipher text component to each unrevoked user to revoke access. The attribute revocation mechanism has a limitation in KP-ABE (key-policy attribute-based encryption) systems.

Homomorphic encryption, multi-cloud computing, and mobile devices have been included in the proposed encryption methods [15] in which the shortcomings of AES (Advanced Encryption Standard), DES (Data Encryption Standard), Blowfish, and RSA (Rivest, Shamir, and Adleman) have been described in detail.

The problem with these systems is that they are unable to strike the right balance between computational overhead and security.

[16] has presented a cloud-based mobile storage system (Android-based phone application) with the aim of integrating cloud storage for all types of mobile data. The proposed solution includes an exact clone of the Android SDK and Java IDE.

Gunjal Yogita [17] the proposed method enables the integration of data fault localization and storage accuracy insurance. The problem with this strategy is that it depends on keys, where key management is secondary.

The size of the key should be proportional to the number of attributes, according to [18]. To solve this problem, an attribute-based cipher text encryption scheme has been proposed. CP-ABE (Cypher text Policy Attribute Based Encryption) is more expensive than KP-ABE (Key-Policy Attribute Based Encryption).

Yang and colleagues proposed multiple authorization levels with attribute revocation in CP-ABE (2014). An efficient administrator was required to maximize the efficiency of the system. Many ABE systems include a revocation procedure.

According to Ibraimi and colleagues (2009), negative remarks in ABE can be used to revoke a user's privileges ABE. When a customer's access is revoked, they are added to a new group of revoked customers. Their strategy is flawed. There is a mechanism for revoking access to the distributed storage server key using a unique encryption key.

[19] Developed another CP-ABE approach based on direct secret sharing for cloud storage services. Yang et al. (2014) developed an extended CP-ABE solution for this method that includes multi-attribute authority and access revocation. However, to perform sophisticated tasks, this system requires a higher level of technical knowledge on the part of the administrator.

3. Access Control in Cloud

Access control is a term that refers to the process of restricting access to certain resources. The term "access" in the context of computer systems can refer to a variety of different things, ranging from the simplest (such as the ability to read, write, share, and delete) to the most complex. Authorization is a term that refers to the ability of an asset to be used.

3.1. Role-based access control (RBAC)

Ferraiolo et. al., introduced role-based access management to this industry in 1995. Sandhu and colleagues also made a significant effort (1996). The proposal includes a core RBAC model, static separation, hierarchical RBAC, and a model for modeling dynamic connections among the various responsible entities [20].

3.2. The Core RBAC model

S, R, OP, and OB denote the subject, roles, objects, and operations described in this section. Any role assigned to an employee by his employer (SA) is always a component of the larger product S and R. It is a

relationship that has been assigned multiple functions based on the subject. Permissions are assigned to a subset of subjects and corresponding roles. PERM can be used to specify both general permissions and their assignment to specific roles:

$$\{perm \in PERM \mid (perm,r) \in PA \}$$

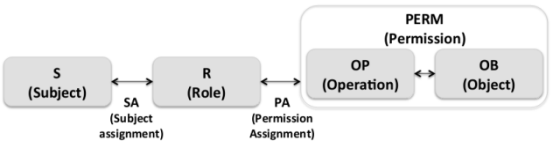

Figure 4. The fundamental modules of RBAC [20]

3.3. NIST RBAC

Because of the shortcomings of the role-based and attribute-based models, Kuhn et al. developed a new approach (2010). They noted a trend toward a role-based paradigm that places a higher value on the access characteristics required to use the system. Before permission is granted, a person must be assigned a position. Before an item can be executed, the subject must be allowed to actively participate. As a result, the system is accessible only to those with the appropriate authority [21].

Authorization: The person has only limited authorization to perform a transaction.

Permission to actively participate. Due to this restriction, users can only use their permissions if they are currently active. Responsibilities are often mixed within the hierarchy. These roles are assigned the permissions associated with the high-level roles. Unlike the DAC paradigm, access is controlled by the system, not the user. MANAC rights management is different from RBAC rights management. Basic operations are controlled by MANAC (such as read, allow user to write & other "security attributes"). With RBAC, control of the system can be as simple as an e-banking transaction or as complex as a read or write operation. This graphic shows management that has more to do with roles than titles. There are numerous methods to approach the RBAC task. The typical RBAC paradigm is shown below, along with an example of an RBAC role hierarchy [22].

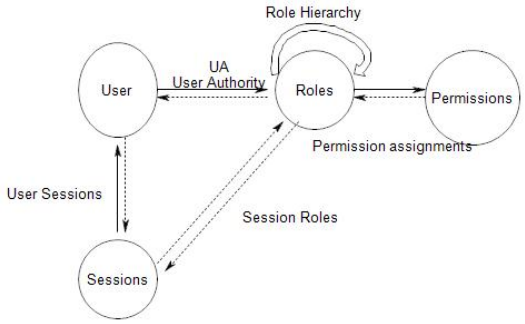

Figure 5. Illustration of Elements in A Standard RBAC Model

3.4. Enhancements in RBAC

As in the time domain RBAC (TRBAC) [23], the role of the requester is assigned based on time constraints, or an extension is made based on the location of the requester. This model is referred to as LBAC, an acronym for location-based access control [24]. In addition, role-based access control has evolved to include a task-based model (T-RBAC) [25] and a privacy-focused approach (P-RBAC) [25]. TRBAC can only be assigned to a specific time period, whether fixed or recurrent. Bertino et al. presented the activation and deactivation of roles with time constraints in the TRBAC paradigm (2001). There are two different states of play: dynamic and static. The first person is able to perform the tasks associated with the subject, while the second person is unable to perform the functions associated with the subject. In addition to roles, [27] discuss GTRBAC, a TRBAC extension that includes temporal filters and user privileges associated with assigned roles.

This is just one example of how location-based access control (LBAC) can be used to meet mobile application and security requirements. In this situation, for example, access is granted to the subject if a particular location or attribute of a location meets certain security requirements. The support role for RBAC

described by [28] is beneficial for a geographically dispersed organization. The attributes of the role, such as highways, cities, or hospitals, help define the boundaries of the role. Before the user can fulfill any of the obligations, he must define the location of the boundary.

[29] Extended the location-based model by adding spatial and temporal filters for objects and customers. Decisions made by this model are influenced by time, which is an enterprise environment, and the duration of business activities with assigned roles and tasks. Sainan pioneered the representation of task-user constraints (2010).

According to [30] who evaluated the TRBAC approach, an intermediate level between a user and permissions should be considered. This was a revised version of [31]. In addition, this work describes requirements and promises as filters for authorizations for use cases. Masoumzadeh's approach assigns roles to clients, distributes objects based on their task functions, associates permissions with causes, and applies permission filters.

3.5. Multi-attribute-based encryption

MA-ABE (Multi-attribute-based encryption authority) is proposed to solve the problem that a single attribute ABE is not available. In general, however, each organization is responsible for a different piece of information. All driver personal information is maintained by the Vehicle Licensing Agency (VLA) and the National Anti-Doping Agency (NADRA).

In order to integrate the information collected by the licensing agency and NADRA, at least one institution must have full confidence to monitor it. If the full confidence of the authority is broken, all data stored in both organizations will be compromised. Multi-attribute models ABE are desired, where one or more mechanisms can act on a request, each issuing its own set of keys with different qualities.

Chase (2007) originally proposed an effective MA-ABE technique. The owner of the information can assign k attributes to each authority in its system, which are monitored by a central authority. The encryption is secure until each authority grants access to a limited number of attributes, and this technique ensures that all attributes are encrypted [32]. On the other hand, the central authority must maintain its integrity while limiting its capabilities. Although the central authority has no attributes and only distributes keys, it is fully trusted. Clients of Chase's model are given a global identifier (GID). This is a special feature of GID:

Each authority has the power to authenticate a user's GID, which means that no client can claim another's. Consequently, the GID can be used to authenticate a user. When two users request the same authority, their GIDs are used to distinguish them.

The Chase unified authority addressed the weaknesses of ABE (2007). Central authority, on the other hand, remained constant for both CP-ABE and KP-ABE. Chase and Chow came up with new ideas in 2009 to improve their method.

By decentralizing the power inherent in the MA-ABE strategy, it is possible to better protect privacy and security. To identify users, the improved architecture of Chase (2007) continues to use GID with all the previously disclosed features. A common key is used by the two identifiers. Each user must provide his or her secret key, which is then combined to form a decryption key. To avoid intrusion attempts, the authorities are not allowed to communicate with each other. GID is currently completely hidden from attribution authorities due to Chase and Chow techniques. Authorities must combine their claims to obtain the decryption key because they have no way to determine a user's identity. To protect the user's privacy and security, pseudo randomness, anonymity, and un-traceability are used. Among other things, they allow access to more complicated structures and a threshold limit for variables in many areas ABE.

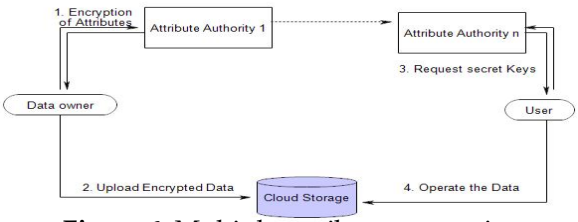

Figure 6. Multiple-attribute encryption

3.6. Analyze the Attribute- and Role-Centered Approaches:

Client authorizations, privilege management, ABAC, and client privilege checking are not included in the above operations. In the attribute-based approach, roles are defined based on the attributes of

individuals. In this way, permissions are not assigned roles, as is the case in the role-based model. This method has a number of disadvantages. In addition to the attribute-based ABAC model, the ABAC models mentioned above can also be used.

The ABE system shares a common property with a large number of users. Under certain circumstances, members of a group can rekeying. Rekeying is a security mechanism that allows another user to view previously encrypted data before it is re-encrypted with the new attributes and keys. On the other hand, a revoked user retains access to the encrypted data until the specified notice period, which is called forward secrecy. This approach was vulnerable due to these two vulnerabilities. This model is used in conjunction with a set of permissions and constraints applied to the objects that can be accessed. A user's basic RBAC provides a set of permissions. This paradigm allows a single session to be granted an unlimited number of permissions. Certain permission sets are associated with specific users. For each set of permissions, a Boolean variable is used to perform a functional mapping from a subset of an object to a set of subjects. This permission is filtered based on sessions and associated subjects. If a function returns false, the user's permissions are revoked and the subject is deleted [33].

The approach of ABAC assumes that whoever controls a particular set of attributes also controls every record within that set. In a distributed storage arrangement, the owner of the information is not solely responsible for the information. During the revocation process, the subject attributes are reactivated. The cipher text is directly linked to the unique identifier property of each client.

In cloud storage solutions, there are no attribute-based methods for access security. We discussed some of the most commonly used attribute-based approaches and other security ideas. The cloud stores remotely managed, maintained and secured data. This network service is accessible via the Internet. The Internet platform allows consumers to store their data and access it from any location [34].

3.7. Layering two access control analysis

As with attribute-based access, layering enables users to choose from a variety of access options during the course of a request. Each time an entity makes a request for access, the layered model verifies the pseudo-role associated with the requesting entity and then compares it to the pseudo-role included in an approach that is relevant to the desired protest. Layered access control checks that the strategy's standards are met by the subject's pseudo-role [35]. If one of the tenets is genuine, access is permitted; otherwise, access is denied. In layered access control, evaluation-access takes the subject, action, and object sought into account and provides a Boolean value for an entrance control decision. For Eval-Access to be regarded authentic, the individual's s, operation, and ob must be members of the datasets S, OP, and OB. When the subject's pseudo-role fulfils a pseudo-role function in the object's policy and one (the access rule) is judged to be real, the access demand via s to execute op over ob is authorized. In the worst-case scenario, Eval-Access is O(|Ru|). The following diagram illustrates the Eval-Access operation [36].

Eval-Access

As the following BOOLEAN expressions demonstrate (ob/op/s-name outcome! :

BOOLEAN. When an object has a satisfied object strategy (ob) and a satisfied pseudo-role (PRF, pr), The pseudo-role is referred to as a "fulfilled" pseudo-role, and the object is referred to as a "successful" pseudo-role.

Proposed Hybrid Access Control Model:

Our goal is to provide a solution that combines the advantages and disadvantages of role-based access management and attribute-based access control. This strategy maximizes the benefits, especially security while minimizing the computational overhead.

The proposed design combines the role-based access management and attribute-based access control to maximize the security while minimizing the computational overhead

The design leverage semantic technologies to incorporate contextual information into role-based models. The context of a user can be conceived as a dynamic role associated with the user. Before other users can access the user's resources, they must be classified using semantic technologies and a description logic (reasoner). In addition, the reasoned can be used to ensure that access rules are consistent. The limited ability of the reasoner to fully explain itself is addressed by using the SPARQL queries instead of formal reasoning. The model as a whole is defined using OWL ontologies. The ideas of role-based access control are outlined in this section using an OWL-DL ontology. Afterward, a relationship is established between

this ontology and the field ontology. And explain, among other things, the responsibilities of the DL classifier and security officers. The next sections provide specifications for owl-expressiveness, the use of SPARQL requests (queries) by dl, and the integration of SPARQL with owl-dl. In developing the role-based ontology for access control, we were guided by three lessons: To maintain simplicity and ease, the ontology for role-based access control must be limited to articulating the modelling concepts of role-based access control; b) when dealing with domain knowledge, synthetic or semantic hypotheses will never be made; and c) when dealing with modelling work-flow processes, ontologies will never adopt a model work-flow process. The following hypotheses are supported by a device that displays role-based access control ideas to be mediated and used via domain ontologies, such as the hospital ontology shown below. This image shows three possible ontologies. The third, fourth, and fifth are domain-based ontologies (user-defined and general).

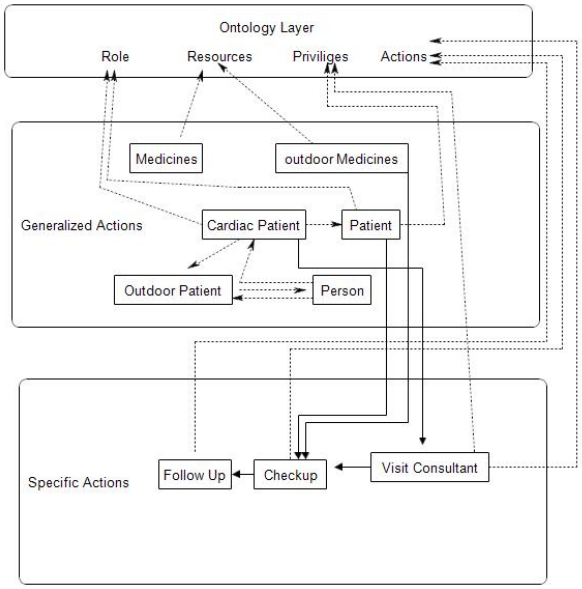

Figure 7. Layering Scheme of RBAC and domain ontologies

There are two actors at work in the organization of role-based access control for all domain ontologies (DL classifier and safety supervisor). Let us assume that the safety supervisor correctly explains first the privileges and second the relationship of the privileges to the modules in the domain ontology. Proverbs originally specified in the domain ontologies using any legal implication, e.g., inheritance, are circulated by the DL classifier.

Each of the four components of the role-based access control ontology looks like this:

When an entity asserts an action, the first module, called "activity," is called a "standalone or incomplete class." It is supposed to serve as the basis for all original actions that can be performed in the system, such as modifying and reading, etc. Adoption is the second module. It is said that chemicals for admission fall into this special category. Privilege is the name for the third module. As a stand-alone module, it illustrates the relationship between resources (Resources) and activities (Activities) (Action). Role is the last and final module. It is classified as well-defined. All modules that meet the hasPrivilege requirement have been classified as belonging to Role by DL. Privilege. So let us assume that domain-based ontologies classify classes by "roles", where each class is supposed to have certain privileges.

The role-based ontology includes the properties listed below:

Many-to-Many Privileges and Roles: This is the relationship between privileges and roles.

The many-to-many relationship of roles with roles, expressing a dynamic separation of responsibilities, is referred to as not-together-with.

Achieves a Goal: Privilege Action is a complete function that associates all of a user's available privileges with the specific actions they can perform as a result of those privileges.

There are several ways to use asserts. Privileged Assert: a comprehensive function that connects to every Privilege an Assert it works with.

The problem with the lack of expressiveness of owl dl's should be addressed here. It is possible that the applicant is referred to as a patient with a rash and is treated by the primary physical fitness physician.

A patient is treated by the applicant. It is possible to alleviate the symptoms of skin rashes. the primary physical fitness doctor protection.

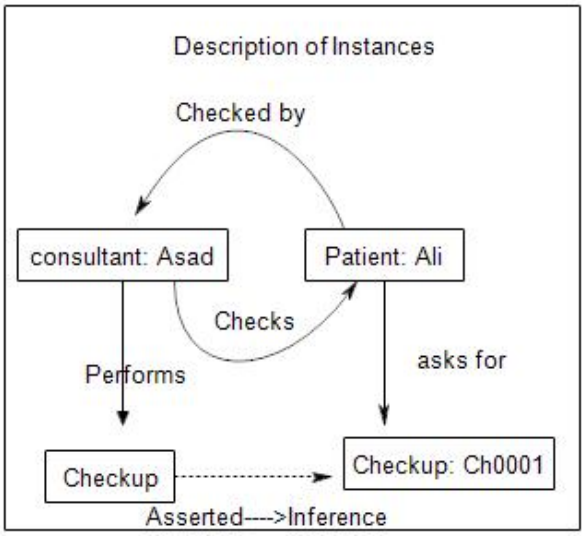

Figure 8. Description of class instances

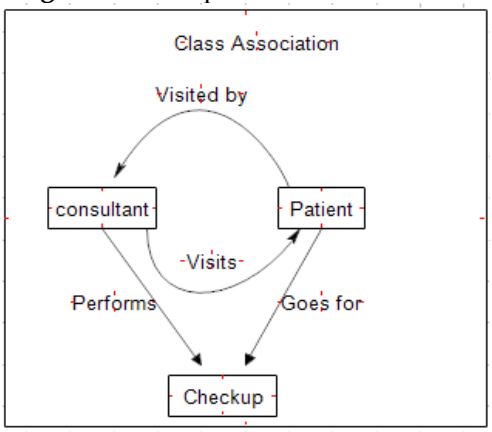

Figure 9. Classes and their Association

In other words, the physician sends the application to the skin department and reads the rash report from the applicant's dermatologist.

Is read by the physician. Sends an application to the specified address. Reduction of skin thickness.

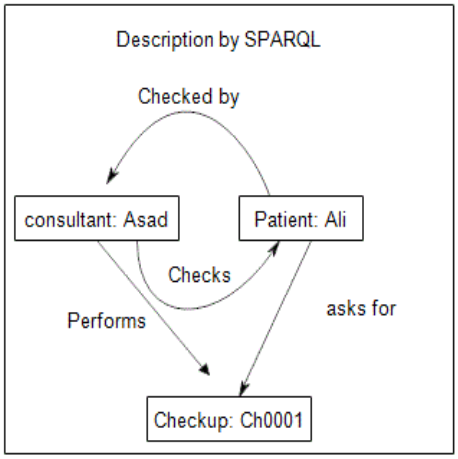

Figure 10. Instances and their description

There is no way to determine if the original element of the rash report generated by a particular physician is the same element of the rash reviewed by the assigned physician, as the above statements simply include classes. To solve this problem, you can use the classifier and the subset of Horn clauses. However, it is possible to reduce the challenge of validating path features between elements to a question

about the existence of this form of path in an ontology. Query statements in sparql provide a solution to a problem.

Following the usual rule for database systems, the language uses a closed-world assumption. This is in contrast to the open-world premise used by DL. This does not imply that certain classes of instances are created to alleviate a particular constraint, as is the case.

ASKconstruct is well suited to our needs because it ensures that few of the constraints identified during design are encountered during implementation. In our search for a match in the graph design, we use the query where-unit to denote the graph unit. Our owl-based ontology is integrated with Sparql queries via annotation properties that convey meta-information about an ontology. Therefore, the annotation properties require false and true are explicitly specified here. Domain-based ontologies can thus also specify additional constraints that are tested in the runtime environment. The string referenced by the property is returned by Sparql queries, which evaluate to true or false.

```
ASK WHERE {
Asad visits ?Ali;
is_advised Consultant.
Consultant checks ?checkup}
```

Figure 11. SPARQL query description

3.8. Simulation Details

This technique uses an agent-based paradigm for deployment and analysis. The technique used to gain access to a cloud environment cannot be replicated with specialized software.

NETLOGO [45] is used as a simulator for this ABAC validation, which is performed in Java. Netlogo is a free and open source integrated platform for financial modelling in cloud environments. A large number of agents are used to simulate a variety of different environments. In addition, this method can be used to analyze the dynamics of a system. Below is a glossary of NETLOGO-specific vocabulary [37].

The term "world states" refers to simulated settings that define the broad environment in which agents must operate. Throughout the simulation, all agent values and state variables are calculated and evaluated.

Breed is a collective noun that refers to a group of agents that are all in the same state at the same time.

Agents can be divided into the following categories.

Turtles are part of an established cosmos that existed before they were born.

Static agents that are a part of the globe patch. Turtles can only interact in areas defined by a static environment that can only be accessed by one turtle at a time [38].

Connections. Turtles have a lot in common. The overwhelming majority of connections are two-dimensional in nature. Types are divided into two categories: directed and undirected.

In this section, we will deal with the observers, which are responsible for monitoring the course of events. These tools allow you to perform evaluation-level tasks and visualize the results in a variety of settings [39].

3.9. Environment Defined in simulation

In sections 3 and 4 of the study, we replicated the health care environment. We invented agents such as doctors, patients, and nurses to help us understand each other's points of view. They are all equal as turtles because they all adhere to the same access control policy, which is set at the level of racial attributes. Access policies are set by observers who have been authorized as cloud administrators. In addition, these permissions can be derived from the rules defined by the administrator. The administrator property defines the properties of agents and provides the user with comprehensive control over these attributes and operations. Link agents, also called link agents, manage data exchange between turtles. A specific network of links exists between nodes. Once the authorization procedure is completed, clients can communicate with each other.

Administrator agents are responsible for creating the policies that govern which users have access to which resources and when. Every profile in the ecosystem is connected to these rules in some way. The observer grants access only after verifying credentials at the attribute level. After that, it is able to establish links. You can access the scenario's analytics using the scenario's behavioral option. The behavioral model of the network logo is an integrated tool that works in conjunction with the logo itself.

Simulation Parameters		
Number of Nodes.	180	
Observer Node, act as cloud operator that control the network	1	
Simulation Model:	Behavior Model	
Simulation Region i.e. size of World	Random i.e. pixel per *x* and *y* coordinates respectively (random-pxcor random-pycor) pxcor and pycor).	
Simulation Device	Intel i5 Core	2.50GHz
	Process cores	2 x 2.50GHz
	RAM	6 GB
	OS	Windows 7 64 bits

Figure 12. Detailed information regarding the simulation parameters for the Net Logo

The following part contains diagrams illustrating the world view of the cloud environment and the scenario for connecting agents (nodes) in the environment [30].

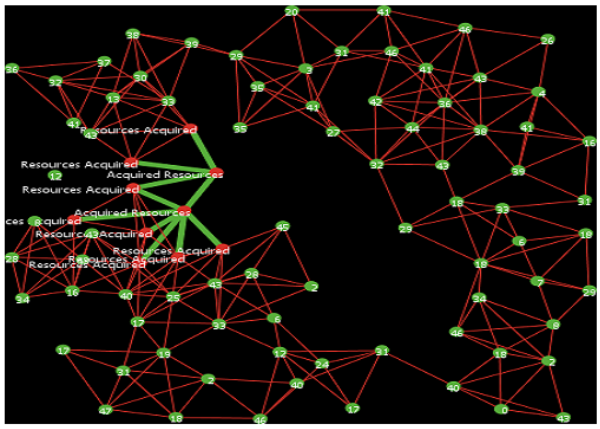

Figure 13. 180 nodes represent the environment

3.10. Evaluation of Parameters

The environment is examined in terms of a total of three factors. This is explained in more detail in the following paragraphs.

3.11. Response Time

System response time is one of the National Institute of Standards and Technology's (NIST) quality performance measures. We track how long it takes consumers to respond to requests to grant access. This time is determined by the number of concurrent client requests. In technical language, network administrator agents, also called observers, are responsible for receiving access requests from system administrator agents, also called turtles. The administrator agents can accept or reject a request based on the information they receive. When a node accepts a request, it establishes a data connection with the recipient of the request. This is identical to what happens in the real world when the resource owner approves or revokes a request. When creating user accounts, we followed the same rules as described in Section 3. An agent authorizes a client's request and the client is then granted access to the system [41].

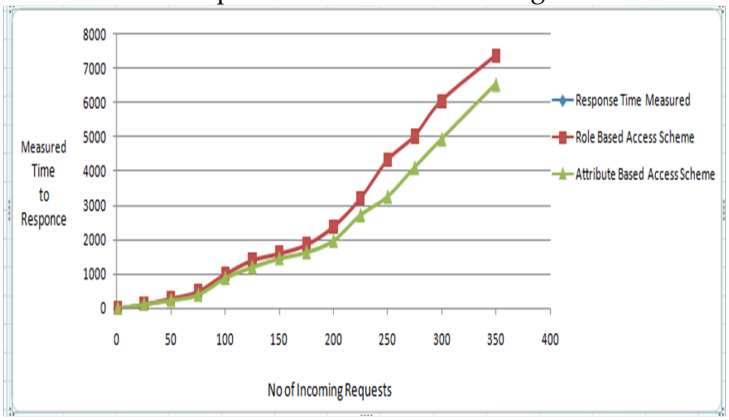

Figure 14. Comparison of Response Times

3.12. Time Required to Launch a New Service

Customers are aware that if the time required to provide a service exceeds the performance of the entire system, they will be dissatisfied. Moreover, it is impossible to handle a large number of requests if the system takes a longer time to provide the requested service. Our method has been validated and proven to have the shortest latency. To be clear, the system does not need to repeat any of the verification steps, which reduces latency.

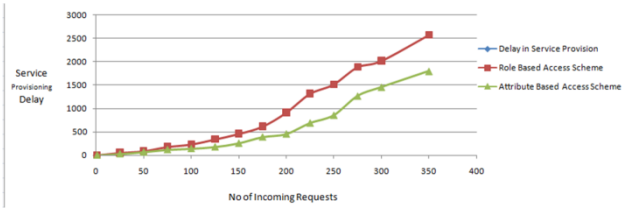

Figure 15. Delay in Service Provisioning

3.13. Communication cost:

During a transmission, the overhead for each data packet is determined. We timed the turtle's interactions with other agents and the administrator node. In today's world, it is quite common to seek and offer services via this type of communication.

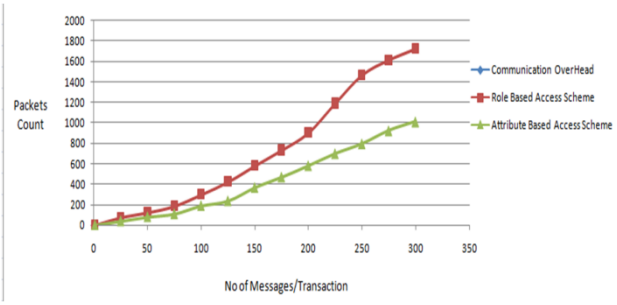

Figure 16. Comparison of the Communication Costs

After you determine the number of requests and messages, the time is calculated in proportion to the number of requests and messages transmitted simultaneously. The overhead increases linearly in direct proportion to the number of requests to send messages. Policy management and efficient delegation of services are responsible for the delays that occur throughout the service provisioning process. Due to the linear increase in time, the availability of the system is large for a large number of applications. Due to the larger management overhead at all levels, the overhead of a traditional system is higher than that of a more modern system. As a result of this single-level visualization technique, the communication costs associated with the attributes are linear.

4. Conclusion

The Layered Access Control paradigm has been studied in depth in this study. Theoretically, access control models were matched and evaluated using an evaluation framework developed specifically for this purpose. The framework was used to analyze the time complexity of a number of access control models, including role-based, attribute-based, and layered access control. The result of these analyses was a comparison between layered access control (LAC) and role-based access control (RBAC). In terms of basic guidance allowed and change of consent, RBAC outperforms layered access control, while in terms of denial, RBAC outperforms layered access control.

There is no doubt that RBAC beats LAC when it comes to approach changes and eliciting client consent. However, compared to attribute-based access control, the display of layered access control outperforms it in terms of approved base guidance and denial, and outperforms it in terms of strategy and consent changes, as well as in the customer authorization survey.

In this work, ACM algorithms (access control model) were explored and analyzed, focusing on efficiency (performance), which was maintained throughout. A second avenue is the design/creation and modeling of frameworks that conform to the model, whereupon their performance can be evaluated with respect to established algorithms (methods) or functions. This includes, among other things, time spent analyzing individual needs and renewing consent. Since there is no consensus on the ideal way to implement these models, it is difficult to defend this strategy when evidence of results is critical. The scope of this dissertation precludes the development of all models with indistinguishable effort parameters required for a reasonable level of correlation accuracy. Because of the time complexity of the material, a hypothetical correlation such as that shown above is the most appropriate technique for this explanation.

References

1. Abderrazak Jemai, S. B. (2015). Enhancing data security in cloud computing using a lightweight cryptographic algorithm. The Eleventh International Conference on Autonomic and Autonomous Systems.
2. Alshehri, S. (2014). Toward effective access control using attributes and pseudoroles. Rochester Institute of Technology.
3. Alshehri, S., Mishra, S., & Raj, R. (2013). Insider threat mitigation and access control in healthcare systems.
4. Arun Kumar, K. K. (2015, March). Cloud-Based Mobile Multimedia to Design a Distributed Recommendation Cache. International Journal of Science and Research (IJSR), 4(3).
5. Balamurugan Balusamy, P. V. (2015). Enhanced Security Framework for Data Integrity Using Third-party Auditing in the Cloud System. 325. New Delhi: Springer.
6. Bertino, E., Bonatti, P. A., & Ferrari, E. (2001). TRBAC: A temporal role-based access control model. ACM Transactions on Information and System Security (TISSEC), 4(3), 191-233.
7. Bonatti, P., & Samarati, P. (2000, November). Regulating service access and information release on the web. In Proceedings of the 7th ACM conference on Computer and communications security (pp. 134-143). ACM.
8. Chase, M. (2007, February). Multi-authority attribute based encryption. In Theory of Cryptography Conference (pp. 515-534). Springer Berlin Heidelberg.
9. Cheng-Kang Chu, S. S.-G. (2014). Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage. IEEE Transactions on Parallel and Distributed Systems, 25(2), 468-477.
10. Damiani, M. L., Bertino, E., Catania, B., & Perlasca, P. (2007). GEO-RBAC: a spatially aware RBAC. ACM Transactions on Information and System Security (TISSEC), 10(1), 2.
11. Enrico Bocchi, M. M. (2014). Cloud Storage Service Benchmarking: Methodologies and Experimentations. IEEE.
12. Everaldo Aguiar, Y. Z. (2014). An Overview of Issues and Recent Developments in Cloud Computing and Storage Security. Springer.
13. Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC), 4(3), 224-274.
14. Hsiao-Ying Lin, W.-G. T. (2012, June). A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 23(6).
15. Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., ... & Scarfone, K. (2013). Guide to attribute based access control (ABAC) definition and considerations (draft). NIST special publication, 800(162).
16. Hussain AlJahdali, A. A. (2014). "Multi-tenancy in cloud computing," in Service Oriented System Engineering (SOSE). (pp. 344-351). IEEE.
17. Ibraimi, L., Petkovic, M., Nikova, S., Hartel, P., & Jonker, W. (2009). Mediated ciphertext-policy attribute-based encryption and its application. In Information security applications (pp. 309-323). Springer Berlin Heidelberg.
18. Joshi, J. B., Bertino, E., Latif, U., & Ghafoor, A. (2005). A generalized temporal role-based access control model. IEEE Transactions on Knowledge and Data Engineering, 17(1), 4-23.
19. K. Sharmila, V. V. (2015, December). Security and Privacy for Storage and Computation in Cloud Computing. International Journal of Science and Research (IJSR), 4(12).
20. Kan Yang, X. J. (2013, September). An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing. IEEE Transactions on Parallel and Distributed Systems, 24(9), 1717-1726.
21. Karun Handa, U. S. (2015, May). Data Security in Cloud Computing using Encryption and Steganography. International Journal of Computer Science and Mobile Computing, 4(5).
22. Keiko Hashizume, D. G.-M. (2013). An analysis of security issues for cloud computing. Journal of Internet Services and Applications.
23. Li, J., Liang, J., Zhao, Q., Li, J., Hong, K., & Zhang, L. (2013). Design of assistive wheelchair system directly steered by human thoughts. International journal of neural systems, 23(03), 1350013.
24. Masoumzadeh, A., & Joshi, J. (2008). PuRBAC: Purpose-aware role-based access control. On the Move to Meaningful Internet Systems: OTM 2008, 1104-1121.
25. Mehdi Sookhak, E. A. (2014, August). A Review on Remote Data Auditing in Single Cloud Server: Taxonomy and Open Issues. Journal of Network and Computer Applications, 43, 121-141.
26. Ming Li, S. Y. (2013, January). Scalable and secure sharing of personal health records in cloud computing using attribute based encryption. IEEE Transactions on Parallel and Distributed Systems, 24(1), 131-143.

27. Moses, T. (2013). eXtensible Access Control Markup Language (XACML) Version 2.0 OASIS Standard Feb. 1, 2005 OASIS Open. Source: http://docs. oasis-open. org/xacml/2.0/access—control-xacml-2.0-core-spec-os. pdf see also http://www. oasis-open. org/committees/tc—home. php.
28. Ni, Q., Bertino, E., Lobo, J., Brodie, C., Karat, C. M., Karat, J., & Trombeta, A. (2010). Privacy-aware role-based access control. ACM Transactions on Information and System Security (TISSEC), 13(3), 24.
29. Rachna Arora, A. P. (2013). Secure User Data in Cloud Computing Using Encryption Algorithms. International Journal of Engineering Research and, 3(4).
30. Ray, I., & Toahchoodee, M. (2007, July). A spatio-temporal role-based access control model. In IFIP Annual Conference on Data and Applications Security and Privacy (pp. 211-226). Springer Berlin Heidelberg.
31. Sainan, L. (2010, June). Task-role-based access control model and its implementation. In Education Technology and Computer (ICETC), 2010 2nd International Conference on (Vol. 3, pp. V3-293). IEEE.
32. Sandhu, R. (2015, April). Attribute-Based Access Control Models and Beyond. In ASIACCS (p. 677).
33. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. Computer, 29(2), 38-47.
34. Sultan Aldossary, W. A. (2016). Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions. (IJACSA) International Journal of Advanced Computer Science and Applications,, 7(4).
35. Turkmen, F., den Hartog, J., Ranise, S., & Zannone, N. (2017). Formal analysis of XACML policies using SMT. Computers & Security, 66, 185-203.
36. V.Malligai, V. K. (2014, March). Cloud Based Mobile Data Storage Application System. International Journal of Advanced Research in Computer Science & Technology, 2(1).
37. van der Linden, H., Kalra, D., Hasman, A., & Talmon, J. (2009). Inter-organizational future proof EHR systems: a review of the security and privacy related issues. International journal of medical informatics, 78(3), 141-160.
38. Vincent, C. H., Ferraiolo, D. F., & Kuhn, D. R. (2006). Assessment of access control systems. Computer Security Division Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD, 20899-8930.
39. Xu, Z., & Martin, K. M. (2012, June). Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on (pp. 844-849). IEEE.
40. Yang, K., & Jia, X. (2014). DAC-MACS: Effective data access control for multi-authority cloud storage systems. In Security for Cloud Storage Systems (pp. 59-83). Springer New York.
41. Yogita Gunjal, P. V. (2013, April). Data Security and Integrity of Cloud Storage in Cloud Computing. International Journal of Innovative Research in Science, Engineering and Technology, 2(4).
42. Young-Ik, E. O. M., Choi, J. H., Jang, H. S., Kim, Y. W., Kang, D. H., & Song, C. H. (2013). U.S. Patent No. 8,387,117. Washington, DC: U.S. Patent and Trademark Office..Yuan, E., & Tong, J. (2005, July). Attributed based access control (ABAC) for web services. In Web Services, 2005. ICWS 2005.
43. Yuan, E., & Tong, J. (2005, July). Attributed based access control (ABAC) for web services. In Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE.
44. Z. Qiao, S. L. (2014). Survey of attribute based encryption. 15th IEEE/ACIS International Conference, (pp. 1-6).
45. Zu, L., Liu, Z., & Li, J. (2014, September). New Ciphertext-Policy Attribute-Based Encryption with Efficient Revocation. In Computer and Information Technology (CIT), 2014 IEEE International Conference on (pp. 281-287). IEEE.