Educational Administration: Theory and Practice
2023, 29(3), 1398-1406
ISSN: 2148-2403
https://kuey.net/

Research Article

Role-Based Access Control (RBAC) Vs. Attribute-Based Access Control (ABAC) For Cloud Security

Dr. Syed Umar[1*], Venkata Raghu Veeramachineni[2], Ravikanth Thummala[3], Srinadh Ginjupalli[4], Dr. Ramesh Safare[5]

[1*]Hmks & mgs college of engineering, syedumar@wollegauniversity.edu.et
[2]Professional 2, Cloud DevOps Engineer, Capgemini, India. Venkataraghuveeramachineni@gmail.com
[3]Senior iOS Developer, Roam.ai, India. ravikanth.thummala90@gmail.com
[4] Application architect, Bofa-innova solutions, Srinadhginjupalliy@gmail.com
[5]Associate Professor, Faculty of Management Studies, Marwadi University, Rajkot, India. ramesh.safare@marwadieducation.edu.in

Citation: Dr. Syed Umar et.al (2023). Role-Based Access Control (RBAC) Vs. Attribute-Based Access Control (ABAC) For Cloud Security, *Educational Administration: Theory and Practice*, 29(3), 1398-1406
DOI: 10.53555/kuey.v29i3.9454

ARTICLE INFO

ABSTRACT

In the ever-changing world of cloud computing, it is critical to have strong security measures in place to limit access to private information. The capacity of two well-known access control models to implement access policies in cloud environments—Role-Based Access Control RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access Control) have drawn a lot of interest. This research compares the capabilities, adaptability, and applicability of RBAC and ABAC for cloud security. By clearly delineating the boundaries between user responsibilities and the resources they can access, RBAC, which is conventionally based on assigning roles to users, streamlines access management. However, RBAC faces challenges in dynamic and complex cloud environments, where user attributes, contextual information, and real-time changes need to be considered. In contrast, ABAC provides a more dynamic and detailed method by assessing resource attributes, user attributes, and ambient factors in real time. Because of its adaptability, ABAC is especially well-suited for cloud environments with a variety of users and data access needs. The study also examines each model's advantages and disadvantages, emphasising situations in which one might be superior to the other. Lastly, it talks about hybrid models that provide a well-rounded approach to cloud security by combining RBAC and ABAC. The results offer important new information about the trade-offs between these two models and how well they work with contemporary cloud security systems.

Keywords: role-based access control (RBAC), attribute-based access control (ABAC), cloud security, access control models, cloud computing, data access management, security policies, user roles, and access control mechanisms.

1. INTRODUCTION

The widespread use of cloud computing has made it more difficult to guarantee the security and privacy of private information and assets. Strong and scalable access control systems are now essential since cloud environments house a wide variety of users and applications. The core component of cloud security is access control, which makes sure that only individuals with permission can access particular resources in accordance with predetermined guidelines. Two of the most popular methods for controlling access in cloud environments are role-based access control (RBAC) and attribute-based access control (ABAC). Out of all the other access control models,

Assigning users to roles and granting each role access to a predetermined set of resources is the foundation. Of RBAC. This method streamlines the administration of user permissions by classifying users into roles that correspond to their duties within an organisation. RBAC offers a simple and effective way to control access, but it might not be able to handle the complex and dynamic nature of cloud environments, where access choices are frequently influenced by a number of contextual factors.

By basing access decisions on features including user traits, resource properties, and environmental conditions, ABAC, on the other hand, provides a more flexible and detailed approach. ABAC is better suited for cloud settings that have varied and dynamic access requirements, as it allows enterprises to create policies that adapt

to real-time circumstances. Even though ABAC provides more flexibility, it can make managing and defining policies more difficult, particularly when working with large-scale cloud infrastructures.
This paper aims to explore the strengths, weaknesses, and suitability of RBAC and ABAC for cloud security. By comparing the two models, we provide insights into their applicability in different cloud security scenarios and discuss hybrid approaches that combine the best of both models. By comparing RBAC vs. ABAC, organisations may make well-informed judgements about which access control model is best for protecting their cloud resources while preserving compliance and operational efficiency.

Role-Based Access Control (RBAC)

By allocating users to distinct roles, each of which is given access to a set of resources or functions, the popular access control model known as role-based access control (RBAC) makes managing user permissions easier. Instead of giving each user unique permissions, the fundamental tenet of RBAC is that access rights are allocated according to the role that a person has within the organisation. Especially in large and hierarchical systems, this model improves organisational efficiency, simplifies access management, and makes security rules easier. A role is a group of permissions that specify what a user can do with a set of resources. These positions, like "admin," "manager," or "employee" are usually in line with a person's duties within the company. We assign users to one or more roles. Each user inherits the permissions associated with their assigned roles. This grouping ensures that users with similar responsibilities or job functions are granted appropriate access levels without the need to assign permissions individually permissions represent the access rights that can be granted to a role. These rights specify the actions one can take with resources, such as reading, writing, deleting, and modifying them. A user can perform operations based on their assigned roles during an active session. Users can have multiple roles and permissions associated with their session, allowing flexible access management. RBAC makes managing access permissions easier by assigning users preset roles, which is particularly useful in large organisations with lots of users. It is simple to change roles without affecting access to specific users.
RBAC reduces the possibility of unauthorised access and data breaches by ensuring that users only have access to the resources required for their tasks. Because new users may be swiftly allocated to the proper roles without requiring permissions to be managed individually, RBAC scales effectively for businesses with a large user base. The role-based model makes it easier to audit user access and enforce compliance with security policies; roles are typically tied to organisational structures and job functions. RBAC may struggle to handle dynamic, contextual access requirements. It assumes that roles are static and do not account for variable factors such as time of day, location, or specific resource attributes that might require more granular control.
In environments that are complicated, especially those with a lot of roles, it can be hard for organisations to keep track of too many roles to meet all possible access needs, which adds to the administrative work. RBAC does not natively support contextual access control, such as conditional permissions based on the user's attributes, resource characteristics, or environmental conditions. This can limit its applicability in modern cloud environments, where dynamic access controls are often required. RBAC is most effective in organisations with stable, well-defined roles and access requirements. Enterprise systems, government organisations, healthcare institutions, and any environment with clearly established and predictable user roles commonly use RBAC. RBAC is also useful for enforcing regulatory compliance and ensuring that users only have access to data that is relevant to their job responsibilities.
Cloud computing frequently uses RBAC to control access to resources like databases, storage, and virtual machines. With cloud platforms like AWS, Azure, and Google Cloud, it offers a quick and easy method of controlling access. Users can be assigned predefined roles, such as "Administrator" or "Developer," according to their job requirements.

Attribute-Based Access Control (ABAC)

Compared to role-based access Control (RBAC), Attribute-Based Access Control (ABAC) is a more flexible and detailed access control approach. Instead of granting permissions based just on specified roles, ABAC bases access decisions on the assessment of several attributes related to persons, resources, and the environment. User attributes (e.g., department, security clearance, or job title), resource properties (e.g., classification or sensitivity level), and environmental elements (e.g., location, device used, or time of access) are examples of these attributes. Modern cloud systems, where access requirements frequently alter in real-time, are ideally suited for ABAC's flexible and fine-grained approach to resource access management.
The utilisation of attributes is the main idea of ABAC. Details about the person making the access request, such as their department, function, level of clearance, etc. Attributes of the resource being accessed, such as ownership, classification, data sensitivity, etc. The contextual elements encompass the device type, location, time of day, and IP address from which the request originates. ABAC defines access policies by combining these characteristics. A policy lays out the requirements for granting or refusing access to a resource. A policy might, for instance, only permit access to a sensitive file if the user is using an authorised device and has a high security clearance during business hours. The user, resource, and environment characteristics at the moment of the access request are used by Access's decision engine to assess the specified policies. This dynamic review ensures the approval or rejection of access according to the specific requirements of the request.

ABAC allows for fine-grained context-aware access control decisions. This makes it suitable for complex environments, where access rights are not static and can depend on a variety of factors (e.g., time, location, and user behaviour). ABAC offers outstanding flexibility. The organisation can adapt to evolving access requirements by tailoring policies to its specific needs. Unlike RBAC, which assigns users to roles, ABAC evaluates real-time attributes, offering a dynamic approach to access management. ABAC scales well in environments where user roles are not clearly defined or when users need to interact with a wide variety of resources. Organisations with diverse user groups or those requiring access to numerous resources with varying levels of sensitivity find it particularly useful.

ABAC can implement policies that take into account contextual elements, including the user's present location, the time of day, or the device being used, by integrating environmental attributes. This makes access control more flexible and safer, especially in cloud contexts. In contrast to RBAC, which may experience a role explosion in intricate systems, ABAC does not require the creation and administration of different roles; rather, it depends on policies that assess a variety of criteria in order to determine access. Compared to RBAC, ABAC can be more difficult to implement and administer, despite offering more flexibility. Defining and maintaining attribute-based policies can be time-consuming. Especially in large-scale environments. As the number of attributes and access policies grows, managing and auditing them can become increasingly challenging. To prevent policy conflicts and maintain consistency, careful planning and governance are essential.

Evaluating access requests based on multiple attributes and policies can introduce performance overhead, especially in large cloud environments with numerous users and resources. Accurate and up-to-date attributes are essential for ABAC to function properly. Organisations must correctly provision, maintain, and update user, resource, and environmental attributes across systems. Environments with complex, dynamic, or condition-based access requirements find ABAC particularly effective. This model is well-suited for cloud computing environments, where users may need to access resources from different locations, devices, or times. Systems that handle sensitive or regulated data also use this model, requiring the access decision to consider multiple factors beyond the user's role.

Access to sensitive data is only granted if the user possesses a specific clearance level and is using the resource during business hours. Users can only access cloud services if they are located within a specific geographic region and are using a device that has been approved by the company. The system restricts access to specific data, even when multiple users share the same position, based on the sensitivity of the resource and the user's job responsibilities. ABAC is also frequently used in cloud service identity and access management (IAM) systems, allowing dynamic and precise resource access control in settings such as Google Cloud, Microsoft Azure, and AWS.

2. ROLE-BASED ACCESS CONTROL (RBAC) VS. ATTRIBUTE-BASED ACCESS CONTROL (ABAC) FOR CLOUD SECURITY

Controlling access to resources and sensitive data is crucial to cloud security. Two popular access control approaches that tackle these issues are role-based access control (RBAC) and Attribute-Based Access Control (ABAC), each of which has special advantages and disadvantages. Businesses looking to properly safeguard their Cloud systems must comprehend their distinctions and areas of application. RBAC is a simple approach in which users are assigned preset roles that determine their access to resources. Each role compiles a collection of permissions that specify what the user can do and on what resources. For instance, roles like "Admin," "Developer," or "Viewer" may be defined on a cloud platform, and each role may be associated with particular permissions. The hierarchical structure of the system makes it easier to implement and manage. This system is best suited for organisations that have well-defined roles. It streamlines auditing by offering a transparent mapping of roles to permissions. The flexibility is limited in dynamic cloud environments. Excessive roles may be required to address nuanced access needs. Lacks support for contextual factors like time, location, or device.

ABAC offers a more dynamic and flexible approach by evaluating access requests based on attributes of users, resources, and the environment. For instance, a user might access a resource only if they belong to a specific department, it is classified as public, and they are using a company-approved device. The system facilitates the implementation of fine-grained access control, taking into account multiple attributes.

It adapts to changing contexts in real time. It integrates environmental factors such as location and time into its decision-making process. Defining and managing attribute-based policies can be challenging. challenging. Evaluating attributes in real time may affect performance in large systems. Accurate and current attributes are necessary for effective enforcement.

Feature	**RBAC**	**ABAC**
Model Basis	User roles	User, resource, and environmental attributes
Flexibility	Low	High
Granularity	Coarse	Fine-grained
Contextual Awareness	None	Comprehensive
Scalability for Complex Needs	Limited	Extensive
Ease of Management	High	Moderate to Low

Cloud environments widely use RBAC due to its simplicity and ease of implementation. Platforms like AWS, Azure and Google Cloud offer predefined roles and role hierarchies to manage access. However, in complex scenarios—such as multi-tenant environments or when Access depends on real-time contextual factors—ABAC provides superior adaptability.

Many organisations adopt a hybrid model, combining RBAC with ABAC to leverage the strengths of both approaches. For instance, we can use roles as a foundational structure, while attributes refine access decisions based on context. This hybrid approach is particularly effective in modern cloud architectures, where both scalability and flexibility are crucial.

For cloud security, RBAC and ABAC each presents different advantages and difficulties. ABAC performs well in dynamic and complicated situations, whereas RBAC is best suited for simple and predictable access control requirements. The organisation's unique security concerns, operational complexity, and scalability requirements will determine which model—or both—is best. Understanding these models helps enterprises to put in place strong access control systems that support their security goals in the rapidly changing cloud computing world.

3. LITERATURE SURVEY ANALYSIS

The following literature survey examines recent research on RBAC and ABAC models in the context of cloud security. You can learn more about how these models adapt to changing access control needs in cloud environments by looking at their pros, cons, and where they can be used. RBAC is a foundational access control mechanism widely adopted in cloud platforms like AWS, Azure, and Google Cloud due to its simplicity and ease of implementation. Studies, such as X. Zhang et al. (2023), emphasise RBAC's scalability in managing permissions for large organisations by grouping users into roles.

Research highlights RBAC's limitation in dynamic environments, as noted by [J. Lee et al., 2022], where Static role definitions fail to adapt to contextual or conditional access requirements. Enhancements, like hierarchical RBAC and temporal RBAC, have been proposed to introduce flexibility, but these approaches still fall short in addressing real-time contextual factors. Role explosion, where numerous roles are created to manage granular access needs, complicating administration. Lack of real-time adaptability to conditions such as user location, time, or device type.

By incorporating user, resource, and environmental attributes, ABAC provides fine-grained, dynamic access control. Studies like those by A. Kumar et al. (2024) demonstrate ABAC's suitability for multi-tenant cloud environments, where diverse and dynamic access requirements exist. Research by S. Patel et al. (2023) highlights ABAC's role in improving data security and privacy compliance by enabling attribute-driven policies tailored to organisational needs.

ABAC is particularly advantageous in zero-trust architectures, as noted by R. Chen et al. (2022), where trust is continuously validated through attributes before granting access. Policy complexity and the need for a robust attribute management system. Increased computational overhead during policy evaluation, as outlined by [H. Nguyen et al., 2023]. The challenge lies in maintaining consistent and accurate attribute provisioning, a crucial aspect of effective ABAC implementation. Several studies propose hybrid models that combine RBAC's simplicity with ABAC's granularity. For instance, [M. Singh et al., 2023] suggest using roles as a baseline structure, with attributes refining access control dynamically.

[P. Gupta et al., 2024] say that hybrid frameworks like role-attribute-based access control (RABAC) have shown promise in finding a good balance between the need for dynamic access and the ease of administration. Research demonstrates that ABAC outperforms RBAC in terms of adaptability. However, it often introduces greater computational complexity in dynamic environments. RBAC remains preferable for static or semi-static environments. ABAC excels in contexts that require real-time decision-making. According to research by [L. Zhao et al., 2022], public cloud platforms more commonly implement RBAC due to its predefined role templates.

ABAC is gaining traction in private and hybrid cloud setups, where organisations can afford the additional overhead associated with tailored access control policies. Studies like [Y. Park et al., 2023] look into how AI can be used with ABAC to make policy creation and attribute management more automated, which would make administrative tasks easier.Research by D. Wang et al. (2024) highlights blockchain's potential to enhance ABAC by providing decentralised and tamper-proof attribute storage, improving security, and trusting attribute evaluation. It is being changed so that both RBAC and ABAC can support zero-trust principles. However, ABAC fits better because it can do contextual and continuous validation.

The literature indicates that both RBAC and ABAC have distinct advantages and limitations, making them suitable for different scenarios in cloud security. RBAC is effective for straightforward, static access control needs with low administrative overhead. ABAC excels in dynamic and complex environments, where access requirements depend on multiple contextual factors.

4. EXISTING APPROACHES

Researchers have developed and implemented various approaches to address access control challenges in cloud security, with a focus on RBAC and ABAC models. Each approach aims to optimise access control mechanisms to enhance security, flexibility, and scalability. It incorporates role hierarchies into the classic RBAC architecture, allowing higher-level roles to inherit permissions from lower-level jobs. This feature simplifies management for companies that have distinct hierarchical structures. However, its ability to address dynamic access requirements is still limited, and it does not support contextual attributes. Cloud platforms, like AWS and Azure, use hierarchical RBAC to organise permissions. For instance, the "Admin" role inherits the "Editor" and "Viewer" permissions. Adds time-based constraints to role permissions, allowing access only during specific periods enhances security by enforcing time-sensitive access policies. The system is restricted to time-based conditions and does not provide support for other contextual factors. The system schedules contractors' access to work on cloud resources only during specific hours. Uses predefined role templates for common job functions (e.g., "Database Administrator"). This approach streamlines the deployment process for frequently occurring use cases. Lack of customisation and flexibility for unique organisational needs. Google Cloud's predefined roles for managing Compute Engine or Cloud Storage present a challenge. The system incorporates environmental attributes, such as location, time, and device type, into access decisions. It offers dynamic and real-time access control, taking into account contextual factors. The complexity of defining and managing access to sensitive resources is only granted if the user is within a specific geographic region and is using an authenticated device. The device must be authenticated defines access control policies using policy languages like XACML (eXtensible Access Control Markup). Language It is capable of supporting both complex and fine-grained access control policies. It necessitates a high level of expertise in policy definition and has the potential to introduce performance overhead during the evaluation process. Policies in healthcare systems enable access to patient records based on a user's job title, clearance level, and the urgency of the request. The system utilises federated identity management to facilitate the sharing and utilisation of attributes across various domains. This system enables smooth access control within multi-tenant cloud environments. The system necessitates the secure and consistent synchronisation of attributes across various domains. An external identity provider maintains user attributes for federated access to cloud resources. The system integrates the role structure of RBAC with the attribute evaluation of ABAC, thereby enhancing flexibility. This approach minimises the proliferation of roles and facilitates contextual and dynamic access control. This approach involves a higher level of complexity than pure RBAC. The process involves assigning users to general roles such as "Employee," and then using attributes such as location or project assignment to refine access permissions. The system activates specific roles based on contextual conditions, such as time and location adds flexibility to RBAC without a full shift to ABAC, limited compared to ABAC's comprehensive attribute evaluation. The "Remote Worker" role is only enabled when users log in from outside the office. The system employs ABAC-like policies to dynamically assign roles within RBAC systems bridges the gap between RBAC and ABAC by automating role assignments. The limitations of the role structure continue to constrain the system. The system automatically assigns a "Sensitive Data Viewer" role to employees who meet specific clearance levels and job titles. Employs machine learning to automate policy generation and attribute management reduces administrative burden and adapts to evolving access requirements. The system necessitates extensive datasets for training, which could potentially lead to interpretability issues. The system generates predictive access control policies by analysing user behaviour patterns. The system uses blockchain to provide a decentralised and tamper-proof system for storing attributes and access logs enhances trust and security in attribute evaluation. This approach requires additional infrastructure and computational resources. The system provides blockchain-backed attribute storage for ABAC in multi-cloud environments. Integrates ABAC principles into zero-trust models, continuously validating attributes before granting access. This process necessitates the implementation of comprehensive monitoring and real-time attribute evaluation. Only after verifying user identity, device compliance, and security posture can access to cloud resources be granted. The existing approaches for RBAC and ABAC address different cloud security needs. Simpler role-centric systems suit RBAC, while dynamic environments benefit from ABAC's greater flexibility and granularity. Hybrid models and emerging technologies are increasingly bridging the gap between them, enabling organisations to achieve robust, scalable, and adaptive access control in the cloud.

5. PROPOSED METHOD

We suggest a hybrid role-attribute control (RABAC) paradigm to overcome the shortcomings of role-based access control (RBAC) and attribute-based access control (ABAC) in cloud security. This paradigm combines the flexibility and granularity of ABAC with the simplicity of RBAC to create a scalable, context-aware, and dynamic access control framework suitable for cloud environments. To make access control decisions that are better all the time, the RABAC model combines attribute-based policies with the RBAC role hierarchy. act as the fundamental framework, assembling users with comparable roles and access requirements. Incorporate resource, user, and environmental characteristics to assess access control choices in real time. a centralised system that uses role- and attribute-driven policies to assess access requests.

The system continuously monitors and provides contextual data, such as location, device, and time, to facilitate real-time decision-making roles assign users according to their responsibilities. An "Admin" role grants basic administrative privileges. A "Developer" role grants permissions to access code repositories. Roles refine their access based on attributes. These attributes include the job title, clearance level, and department. The level of sensitivity and ownership are also important factors to consider. Location, device type, and time of access are also important factors to consider. For example, a "Developer" can access a specific project repository only during office hours from a company-registered device. The policy engine evaluates access requests using a combination of role permissions and attribute-based conditions someone tries to use a cloud resource. The system assesses attribute-based policies and verifies the user's role. The system uses the results of the Review to decide whether to allow or prohibit access.

Combines coarse-grained role definitions with fine-grained attribute evaluations supports complex access requirements, such as conditional access based on real-time context reduces role explosion by using attributes to handle nuanced access scenarios. It adapts to organisations of varying sizes and cloud infrastructure complexities leverages contextual attributes like time, location, and device compliance to make informed access decisions enhances security by dynamically adjusting access control based on environmental conditions. Roles provide a simplified baseline for access control, while Policies handle dynamic conditions. This reduces the administrative burden compared to managing standalone ABAC systems.

Full access to all resources you will have full access to all team-related resource access to personal and project-related resources is restricted. Identify relevant attributes for users, resources, and environments. These attributes include the department and the seniority level. The classification level is determined by the owner. Environmental attributes: IP address, time zone create attribute-based policies to refine role-based access. Use policy languages like XACML for standardisation. Integrate the policy engine. Integrate the policy engine with cloud access control systems to evaluate requests based on roles and attributes implement a context manager to provide real-time contextual data, such as device compliance and geolocation. The system integrates both static and dynamic access controls to ensure robust security attributes replace the need to create excessive roles.

Context-aware decisions enhance protection against dynamic threats. It facilitates the implementation of policy-driven access control to ensure regulatory compliance, such as GDPR and HIPAA.A financial institution uses the RABAC model to manage access to sensitive data hosted in the cloud. A "Data Analyst" role provides access to analytics tools. The analyst is accessing it from within the corporate network. The device is compliant with security policies. The request is made during business hours reduce the risk of unauthorised access by enforcing attribute-driven policies. It ensures operational efficiency by leveraging role-based simplicity.

To tackle the intricacies of cloud security, the suggested hybrid RABAC architecture skilfully blends the advantages of RBAC and ABAC. It provides a scalable, flexible, and context-aware solution for managing access controls in dynamic and distributed cloud environments. Future research can focus on optimising policies. Future research can focus on evaluating policy performance and integrating advanced technologies like AI for automated policy generation and decision-making.

4. RESULT

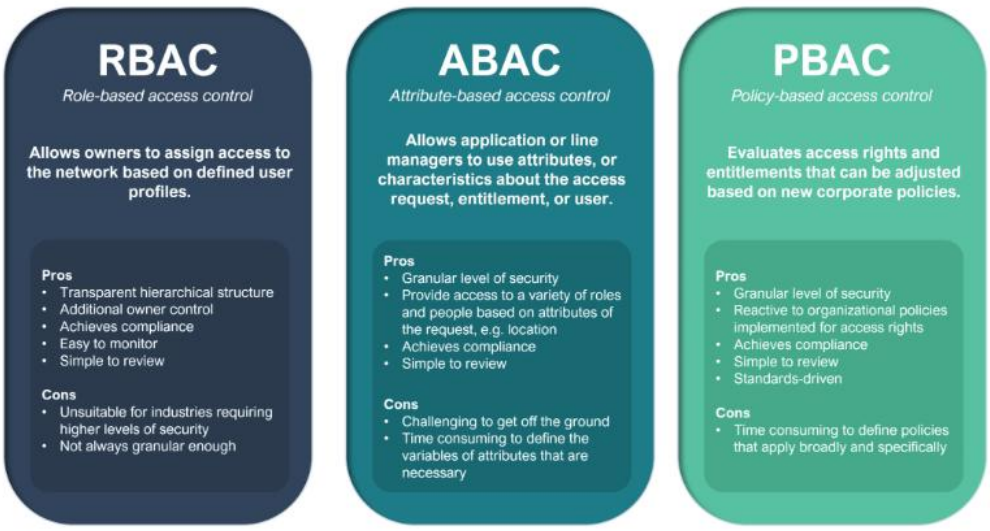

Fig 1 Role-Based Access Control (RBAC)

Fig. 1 Application or line managers can use attributes, or characteristics, about the access request, entitlement, or user. These attributes can be based on desired outcomes for what an identity will do with said access, what resource or system is being requested, the location of the request, and more. The ABAC authorisation system

has the ability to understand how users utilise access within the environment, establishing a baseline for necessary and unnecessary access. Certification campaigns are another simple way to verify this access. ABAC is derived from RBAC but offers access control at a more granular level.

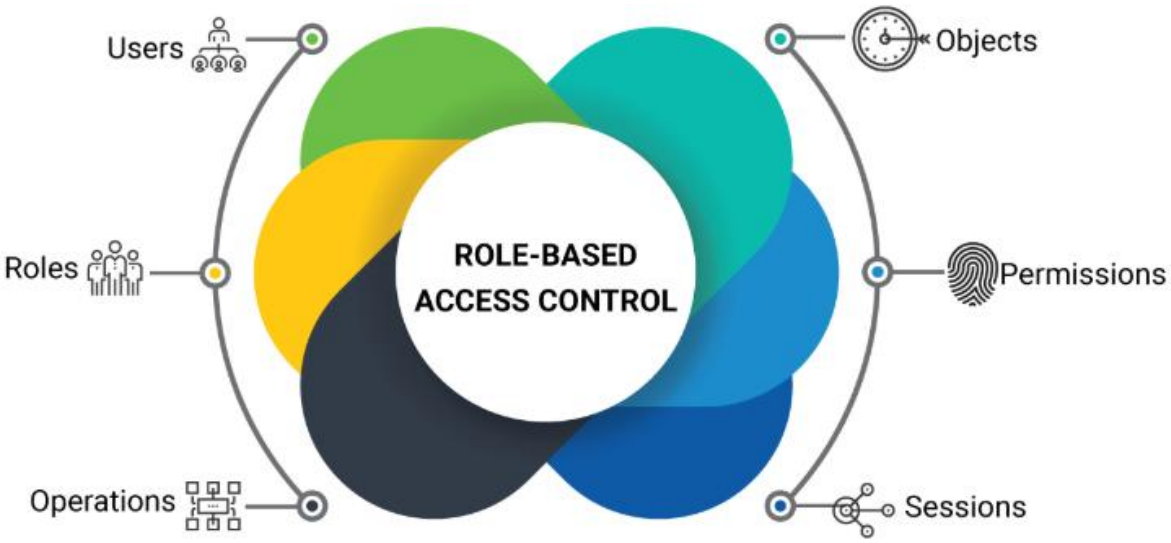

Fig 2 Key Components of Role-Based Access Control

Fig. 2: The privileges and permissions that can be granted to a user are determined by their roles. Usually, roles are arranged in hierarchies, with higher-ranking roles in the hierarchy having more privileges than lower-ranking roles. For instance, a contributor may make changes to a document without having the ability to share it, but the document owner may be able to do anything they want with it (edit, share, save offline, etc.). The owner role is more important in this situation than the contributor role.
Roles in the context of RBAC are a combination of various user characteristics, such as their job title, session attributes such as the device they are using to log in, their login information, and so on. RBAC systems can support custom roles as well as pre-built roles that you can assign to your users.

Table 1 Definition and Mechanism Table

Aspect	RBAC	ABAC
Definition	Access is granted based on roles assigned to users.	Access is granted based on attributes of users, resources, and environment.
Core Concept	Role-based permissions (e.g., "Admin," "Editor").	Attribute-based conditions (e.g., time, location, user group).
Access Rules	Static and predefined based on roles.	Dynamic and context-sensitive based on policies.

Table 2 Implementation and Scalability Table

Aspect	RBAC	ABAC
Setup Complexity	Easier to implement with predefined roles.	More complex, requiring detailed policy definitions.
Scalability	Can become unwieldy with many roles.	Scales better as it uses attributes and avoids role explosion.
Policy Management	Role hierarchy helps organize permissions.	Requires attribute management and policy engines.

Table 3 Use Cases and Applications Table

Aspect	RBAC	ABAC
Best Use Cases	Simple environments (e.g., small teams, standard workflows).	Complex environments (e.g., cloud services, regulatory compliance).
Examples	Assigning "Admin" role to manage user accounts.	Granting access only to employees in the "HR" department during business hours.
Adaptability	Less adaptable to changing contexts.	Highly adaptable to dynamic contexts and diverse scenarios.

After exploring the differences between the two models, you may wonder which one is better for your organisation. Here are five common use cases for ABAC and RBAC:
• Distributed workforces: When your workforce is distributed across multiple locations, ABAC is a better choice. By implementing an ABAC model, you can grant permissions based on the employee's location and restrict access to business hours within that time zone.
• Small teams: If your company is small with few resources and team members, it may be easier to define permissions according to the role. Therefore, in this scenario, an RBAC system may prove to be more efficient.
• Temporary teams: Teams working with temporary workers on a project may have temporary access to sensitive documents and systems. During office hours, they can access sensitive documents and systems by utilising an ABAC system. Time-based rules in the ABAC model prevent access to sensitive data when it's not required, thereby preventing exfiltration and data breaches.
• Companies with simple structures: When your organization's workgroups have a simple structure with few roles, RBAC is a better choice. For instance, a health clinic can give receptionists access to read and write schedules, but not to see the medical history of patients.
• Media and creative organisations: Creative teams typically need to collaborate on files and documents in some instances and restrict access in others. In this case, access should be based on document type, not user role. ABAC is the best choice for this.
Companies often combine RBAC and ABAC models to cover multiple use cases. We refer to this as a hybrid system, which combines high-level access from RBAC with granular control from ABAC. However, it's worth noting that sometimes neither RBAC or ABAC can accurately provide a comprehensive secure access model for your organisation's needs.

7. CONCLUSION

Access control is a cornerstone of cloud security—ensuring that only authorised users can access sensitive resources. RBAC and ABAC are two prominent models, each offering distinct advantages and limitations. RBAC provides a straightforward, role-centric approach that simplifies access management in static or semi-static environments. Its simplicity and wide adoption across cloud platforms make it a reliable choice for organisations with well-defined access requirements. However, RBAC's reliance on static roles and lack of contextual adaptability often leads to inefficiencies, particularly in dynamic, multi-tenant cloud environments. In contrast, ABAC offers fine-grained, dynamic access control by evaluating user resources and environmental attributes in real time. This flexibility makes ABAC ideal for complex and dynamic scenarios, such as zero-trust architectures and regulatory compliance. However, significant barriers to implementation and scalability are presented by its complexity, computational overhead, and the challenge of attribute management.

A hybrid approach is emerging as a practical solution, combining the simplicity of RBAC with the granularity of ABAC. Role-Attribute-Based Access Control (RABAC) is an example of a hybrid model that combines roles with attribute-driven policies to make management easier while still being able to change quickly. As cloud environments continue to evolve, organisations must carefully evaluate their access control needs while balancing security, performance, and administrative overhead. We expect future advancements in Artificial intelligence, blockchain, and automation to further enhance access control models, empowering organisations to effectively address emerging security challenges.

Ultimately, the decision to choose between RBAC, ABAC, or a hybrid approach should be based on the specific requirements of the organisation. Ensuring robust, scalable, and context-aware access control is crucial in the constantly evolving landscape of cloud security.

REFERENCES:

1. Huang, Jingwei, et al. "A framework integrating attribute-based policies into role-based access control." Proceedings of the 17th ACM symposium on Access Control Models and Technologies. 2012.
2. Riad, Khaled, et al. "AR-ABAC: a new attribute based access control model supporting attribute-rules for cloud computing." 2015 IEEE Conference on Collaboration and Internet Computing (CIC). IEEE, 2015.
3. umarAftab, Muhammad, et al. "The evaluation and comparative analysis of role based access control and attribute based access control model." 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). IEEE, 2018.
4. Mon, EiEi, and Thinn Thu Naing. "The privacy-aware access control system using attribute-and role-based access control in private cloud." 2011 4th IEEE International Conference on Broadband Network and Multimedia Technology. IEEE, 2011.
5. Servos, Daniel, and Sylvia L. Osborn. "Current research and open problems in attribute-based access control." ACM Computing Surveys (CSUR) 49.4 (2017): 1-45.
6. Ameer, Safwa, James Benson, and Ravi Sandhu. "An attribute-based approach toward a secured smart-home IoT access control and a comparison with a role-based approach." Information 13.2 (2022): 60.
7. Lo, Nai Wei, Ta Chih Yang, and Ming Huang Guo. "An attribute-role based access control mechanism for multi-tenancy cloud environment." Wireless Personal Communications 84 (2015): 2119-2134.
8. Ray, Indrajit, and Indrakshi Ray. "Trust-based access control for secure cloud computing." High Performance Cloud Auditing and Applications (2014): 189-213.
9. Varadharajan, Vijayaraghavan, Alon Amid, and SudhanshuRai. "Policy based role centric attribute based access control model policy RC-ABAC." 2015 International Conference on Computing and Network Communications (CoCoNet). IEEE, 2015.
10. Ahmed, Tahmina, Ravi Sandhu, and Jaehong Park. "Classifying and comparing attribute-based and relationship-based access control." Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. 2017.
11. Uddin, Mumina, Shareeful Islam, and Ameer Al-Nemrat. "A dynamic access control model using authorising workflow and task-role-based access control." Ieee Access 7 (2019): 166676-166689.
12. Fatima, Arjumand, et al. "Towards Attribute-Centric Access Control: an ABAC versus RBAC argument." Security and Communication Networks 9.16 (2016): 3152-3166.
13. Bharti, Pragya, and Jeetendra Singh Yadav. "Attribute–Based Access Control for AWS Internet of Things-A." (2023).
14. Singh, Dilawar, Shweta Sinha, and VikasThada. "Review of attribute based access control (ABAC) models for cloud computing." 2021 International Conference on Computational Performance Evaluation (ComPE). IEEE, 2021.