Received: 26 February 2019 | Revised: 16 June 2019 | Accepted: 15 July 2019

DOI: 10.1002/ett.3720

RESEARCH ARTICLE

WILEY

A survey on access control mechanisms for cloud computing

Rayane El Sibai[1] | **Nader Gemayel**[2] | **Jacques Bou Abdo**[2] | **Jacques Demerjian**[3]

[1]Faculty of Engineering University, Al Maaref University, Beirut, Lebanon

[2]Faculty of Natural and Applied Sciences, Notre Dame University, Deir El Kamar, Lebanon

[3]LaRRIS, Faculty of Sciences, Lebanese University, Fanar, Lebanon

Correspondence
Rayane El Sibai, Faculty of Engineering, Al Maaref University, Beirut 2827, Lebanon.
Email: rayane.elsibai@mu.edu.lb

Abstract

Cloud computing is an Internet-based computing where the information technology resources are provided to end users following their request. With this technology, users and businesses can access programs, storage, and application development platforms through the Internet and via the services offered by the cloud service providers (CSPs). One of the biggest obstructions in the cloud computing environment is data security. Actually, the data are dispersed across multiple machines and storage devices such as servers, computers, and various mobile devices. The uncontrolled access to these resources and data leads to many important data security risks for the end users. In this way, and in order to ensure the reliability of the cloud and the trust of the users regarding this environment, controlling access to data and resources as well as protecting and ensuring their security becomes a critical task for CSPs. In this work, we present a comprehensive review of existing access control mechanisms used in the cloud computing environment. The advantages and disadvantages of each of these models are discussed and presented along with their analysis. Also, we study the cloud requirements of these models, and we evaluate existing control mechanisms against these requirements.

1 | INTRODUCTION

Cloud computing can save time and money, but it is more important to trust the system. One of the biggest obstructions in the cloud environment is data security. In fact, the users' data are dispersed across multiple machines and storage devices such as servers and computers, and various mobile devices such as wireless sensor networks and smartphones. This makes the security of the data quite serious. The lack of security control when the cloud service provider (CSP) does not provide adequate data protection models will lead to information leakage. This would result in significant loss to the user and will often lead to security risks and system failure.

Nowadays, cloud systems are not reliable enough as claimed by the CSPs. Several cloud security issues have been reported in recent years. One can cite the shutdown of Amazon storage service, which took place twice in 2009. This accident resulted in the shutdown of several network sites that relied on this storage service. Also, in March 2009, security flaws in Google Docs led to serious leakage of users personal information and Gmail shut down for four hours.[1]

The trust in a cloud environment depends on the deployment model and on the data protection and prevention techniques used. In a public cloud deployment, the control of the data access is delegated to the organization owning the infrastructure who is responsible to define a security policy. Rawashdeh et al[2] presented a survey on existing cloud trust and reputation models. A comparison between them was conducted based on the cloud customer feedback.

Data security is the combination of the integrity, anonymity, and confidentiality of the data in the cloud. Data integrity preservation intends to protect the data from unauthorized deletion, modification, or fabrication. It is achieved using database constraints and transactions accomplished by a database management system. When the private data of the users are stored in the cloud, the confidentiality of these data becomes essential to increase the reliability of the cloud.

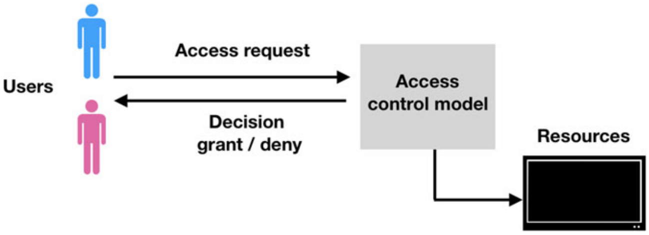

FIGURE 1 Access control mechanism scenario

Data confidentiality can be provided by authentication and access control policies, data encryption, and data storage distribution.[3] To ensure the reliability of the cloud and the trust of the users regarding this environment, it is necessary to protect and ensure data security. Therefore, access control mechanisms should be maintained.

Access control is defined as the restriction of access to a specific place or resource. It is a set of conditions that determine the ability of a person to access data or resources. Access control mechanisms are important because of the services heterogeneity and the dynamicity of the cloud. Access control mechanisms are used to ensure that every attempt of particular users to the object are based on the access privileges given by the system. Figure 1 depicts an access control mechanism scenario.

Cloud-based systems should ensure a maximum level of integrity, scalability, privacy, and availability to their users. However, designing and implementing an access control mechanism is a very complex and critical task for cloud computing systems. In fact, these systems require a set of well-designed and tested security rules, and they pose many problems regarding data access controls mechanisms. One can cite the following[4-6]:

- In cloud computing, data are stored in different locations at the same time. The data owner does not have full control over his resources. Thus, the CSP has the ability to recover the data even if the data owner has deleted its data from the cloud server.
- On the cloud server, user data are subject to internal and external attacks. On the one hand, in the same organization, malicious users can damage other users data since the cloud resources are shared between them. On the other hand, a software bug or hardware failure may lead to data security breaches.
- User revocation is one of the main issues to be addressed when implementing an access control scheme for cloud computing. Actually, when a user is revoked from the system, the user must no longer have access to the data.

In general, our contributions in this paper are as follows. At first, a survey of existing access control mechanisms especially for cloud-based systems is presented. Secondly, an analysis of the advantages and disadvantages of the presented models is performed. Finally, a study for the requirements for access control mechanisms in cloud computing is performed, and an evaluation of existing access control mechanisms against these requirements is carried out.

The remainder of the paper is organized as follows:

- Section 2 introduces the basic concepts of cloud computing. In particular, Section 2.1 presents the cloud services, and Section 2.2 discusses the cloud deployments models.
- Section 3 presents a review of existing access control mechanisms for cloud systems. In details, Section 3 presents the following:
  - Section 3.1 exhibits two languages used to implement access control policies.
  - Sections 3.2, 3.3, 3.4, 3.5, and 3.6 address the discretionary access control (DAC), mandatory access control (MAC), role-based access control (RBAC), attribute-based access control (ABAC), and attribute-based encryption models, respectively.
  - Section 3.7 discusses the federated identity management model and its architecture.
  - Section 3.8 compares all the presented access control mechanisms according to the cloud system requirements.
- Section 4 ends the paper and presents future research directions.

2 | CLOUD COMPUTING

Cloud computing, known as on-demand service, is Internet-based computing where information technology (IT) resources and software applications are provided to computers and mobile devices on-demand. Its main concept is: “Why would you buy anything when you can rent it?". Thus, instead of investing in infrastructure, users and businesses may find it useful to rent the infrastructure and the needed software to run their applications. In this environment, there are service providers (SPs) that facilitate, manage, and render the services to the users and businesses who in their turn will pay the costs of the leased services. Cloud computing provides two basic functions as services: computing and storage. With this technology, the users and businesses can access programs, storage, and application development platforms through the Internet and via the services offered by the cloud computing providers.[7]

The adoption of the cloud environment has several benefits. It allows users and businesses to save time and costs. In fact, companies that manage their own platforms by themselves must buy and maintain their hardware and software infrastructures. This requires human resources with professional knowledge and special skills to take care of the platforms. With the use of cloud computing, the cost of storage has dropped dramatically and the efforts of the infrastructure installation, configuration, and maintenance have been overlooked. In addition, the estimation and planning of the required resources, as well as the use of excessive storage and computation capacities solely to manage maximum workloads are no longer required as the resources can be flexibly adjusted as needed.

2.1 | Cloud service types

Cloud computing provides three types of services to deliver a service to end users.

- Software as a service (SaaS): It represents the capability provided to the cloud users to use and to run applications on the cloud. These applications are accessible by the users through a web interface. The primary focus of the SaaS model is to control the users' access to the applications.[8]
- Platform as a service (PaaS): It is a development platform that enables the users of the cloud to develop services and applications directly on the cloud. An example of PaaS is Google App Engine. A PaaS model differs from traditional applications where users' data are stored locally and are subject to the access control policies defined by the user. In a PaaS model, the data are stored at the PaaS provider end, and the users rely on it for security measures. The major security problem in the PaaS model concerns the network intrusion. It is the PaaS provider who has to deal with this issue and to protect the data from breaches. Thus, efficient data encryption and decryption algorithms, as well as fine-grained authorization techniques must be implemented.[8]
- Infrastructure as a service (IaaS): It consists of providing the cloud users several computing, storage, and network resources using which the users can run their own applications or software. The main concern in this model is the security of virtual machines.[8]

2.2 | Cloud deployment models

Cloud deployment models can be categorized into four types: public cloud, private cloud, hybrid cloud, and community cloud.[9,10]

- Public: It is the popular cloud deployment model. With the public cloud, the SP is the owner of the cloud, and anyone can access its services through web interfaces. Access to services is paid and only for the duration during which the services are used. Many common clouds adopt the public deployment mode, such as Amazon EC2, S3, and Google App Engine.[11] Public cloud is highly flexible, reliable, and scalable; however, it is less secured: among all the cloud deployment models, the public cloud poses the major security issues.[8]
- Private: The private cloud can be compared with an intranet, which is owned by the company in which only the authorized users can access the services provided. Unlike the public cloud where the resources and applications are managed by the cloud provider, the services in the private cloud are managed by the organization itself. The main benefit of this model is its high-security level, particularly, data confidentiality.[12,13] Unlike the public cloud, private cloud is highly secured and has more control over its resources as these later can only be accessed within the organization. Nevertheless, these benefits make this cloud model less scalable and more expensive.
- Hybrid: The hybrid cloud is a composition of several clouds whose infrastructures are distinct. It is the mix of public and private clouds. Notice that the fact of using two types of cloud (public and private) at the same time cannot be considered a hybrid cloud. In fact, clouds must be used in conjunction with each other. For instance, an organization can use a public cloud that processes the data and sends it to a private cloud for storage. In such a case, the cloud is considered a hybrid. This cloud model has the advantage of being flexible, scalable, and more secure than the public cloud.[12,13]

- Community: A cloud of type community is a collaborative cloud computing solution targeted to a limited subset of individuals or organizations. This shared cloud is governed and managed commonly by all the participating organizations or by a third party. This type of cloud is usually used by organizations working on joint projects or research and requiring a shared platform for managing and executing their projects.[12,13] The main advantage of this model is its ability to grow by allowing new users to collaborate.

3 | ACCESS CONTROL MECHANISMS

A complete and effective access control model should ensure the following security requirements[14]: confidentiality, integrity, and availability of the data and resources. To accomplish these requirements, an access control model must provide the following features: authentication, authorization, and accountability.[15]

- Authentication: Username and password remain the most common forms of user authentication credentials managed by identity and access management (IAM) systems, which also support other identification methods, such as digital signatures and certificates, and biometric hardware.
- Authorization: It is the decision of the access control model of granting or denying permission to an authenticated user to access a specific resource. Authorization policies are Boolean functions that are assessed for every access decision. For instance, in banking systems, policies define who is allowed to view, edit, delete, and approve banking transactions. An instance of positive policy would be as follows: A manager can view banking financial transactions. An instance of a negative policy would be as follows: No person can approve a banking financial transaction above their approval level. In large enterprises, policies may be combined to achieve any relevant authorization scenarios.
- Accountability: It represents the ability of the access control model to trace the users' activities.

3.1 | Access control language

The implementation of access control policies requires the use of a specific access control language. One can cite the following two languages:

- Security Assertion Markup Language (SAML): it is an XML-based framework[16] that provides protocols to define the communication sequence during request and response messages. SAML allows the end users to access several services while authenticating only once. Figure 2 shows the SAML system sequence diagram between clients, identity providers (IdPs), and SPs. Clients are the end-users, IdPs can be any third-party entities with identity databases, and SPs are the service providers willing to rent or sell their cloud services, such as SaaS in this case. At first, the user logs in to IBM.com (SP) where SaaS services are available to buy. IBM.com does not manage the authentication itself. It needs to authenticate the user. Hence, it constructs a SAML authentication request, signs it (optional encryption), and finally,

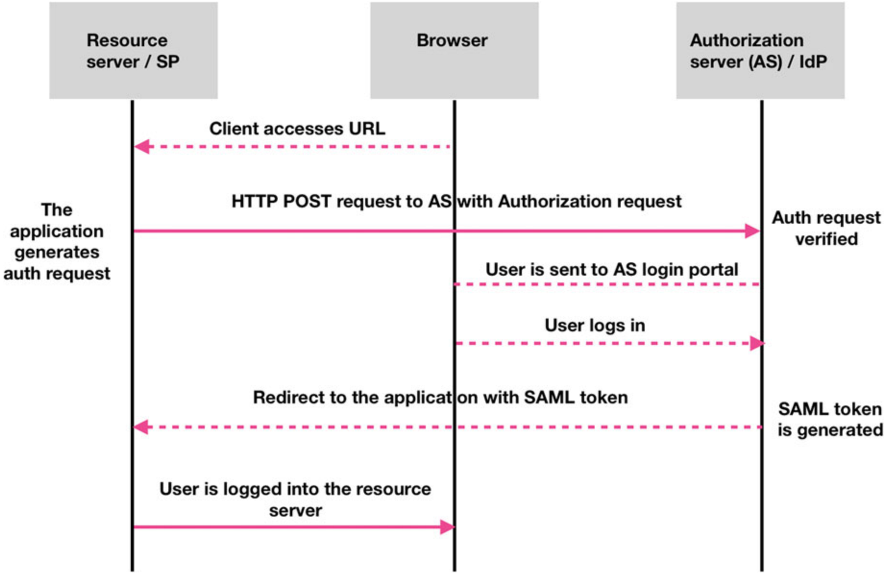
Resource server / SP
Browser
Authorization server (AS) / IdP
Client accesses URL
The application generates auth request
HTTP POST request to AS with Authorization request
Auth request verified
User is sent to AS login portal
User logs in
Redirect to the application with SAML token
SAML token is generated
User is logged into the resource server

FIGURE 2 Security Assertion Markup Language system sequence diagram. SP, service provider

encodes it. Thereafter, the user's web browser is redirected to the IdP for authentication. When the IdP receives the request, it decodes it, decrypts it if necessary, and verifies the attached signature. In turn, IBM.com receives the SAML token and verifies it, decrypts it if necessary, and extracts the user's identity information such as userID and its permissions. The user now might log to IBM.com and perform any desired task. The IdP, in this case, does not hold the user's credentials.

- eXtensible Access Control Markup Language (XACML): It is an easy and adaptable way to establish authorization policies in complicated and active environments. It is also considered a policy language and a request protocol to control decisions. When a user requests to access a resource protected by policy enforcement point (PEP), the PEP assigns an XACML request to the policy decision point (PDP) to verify if the user should or not be given the access. Then, PDP issues a response to the PEP and informs it its decision: permit, deny, intermediate, or not applicable. Finally, the decision is applied by the PEP. XACML has many advantages. Previously, control policies were inputted in different languages by IT agents. Using XACML, the access control policy is written once and used for many applications. In addition, XACML language permits the authorization centralization allowing the access control policies to be handled centrally. To deal with the interoperability issue between different access control models, Hu et al[17] proposed an ontology-based language, called Semantic Access Control Policy Language (SACPL) for cloud computing systems.

3.2 | Discretionary access control

DAC also called Identity-Based access control (IBAC), is defined by the Trusted Computer System Criteria Evaluation (TCSEC). It consists of identifying the user with the credentials provided during authentication, such as the user name and password. In the DAC model, the user has the complete authority over all the resources he owns, and he also determines the permissions for other users who have those resources. DAC mode allows a user with some access permission to pass this permission to any other user. Permissions are the benefits that a user can hang on objects. They allow the user to access an object in a specific mode, for example, read or write. With the DAC model, users access rights are determined at the base of an access matrix, as shown in Figure 3.

In the DAC model, the owner of the object assigns access authorities to users and defines their privileges. Permissions may be granted to a group of users instead of a single user. Thus, the owner of an object creates a group of users and gives them certain permissions and limitations. This group will be controlled by the owner of this group, thus allowing a high level of delegation of administrative capabilities. However, this mechanism can lead to serious security problems if the owner of the group is not trustworthy.[18] For instance, the owner of a group may change the DAC security policy by using malicious software, such as Trojan horse.

Security management in the DAC model is not obvious. The main drawback of the DAC model is its lack of control over the flow of information. DAC is inherent to safety problems due to the lack of constraints and copy privileges. This lack of copy privilege prevents the verification of information. Therefore, the information can be copied from one object to another, allowing an unauthorized user to access a copy of the information even if the object owner has not allowed the user to access to the original information.

3.3 | Mandatory access control

In the DAC model, it is the user, the owner of the object, that sets the access permissions, and that allows or denies other users to access its resources. On the contrary, access permissions to an object in the MAC[19] model are defined by the system

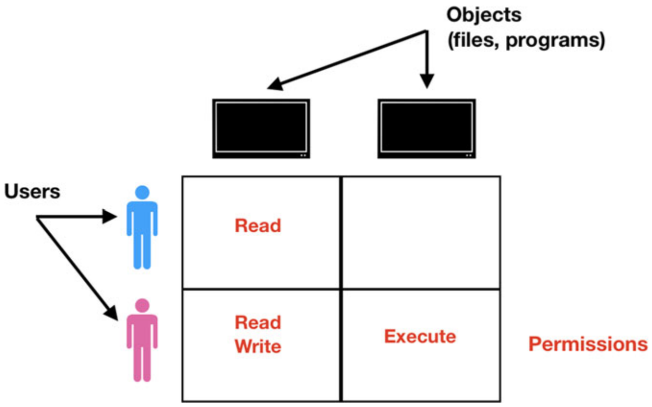

FIGURE 3 Access matrix model of discretionary access control

administrator and the individual users cannot modify these permissions. The MAC model is a security policy owned by the system administrator and not the data owner. The system administrator is the only responsible for managing and defining the access control policy and grating/revoking access rights to users, and these permissions are imposed by the operating system (OS). Therefore, the users cannot alter their rights by granting themselves a higher level of permissions than that defined by the system administrator. In the MAC model, each user and object has a trust and sensitivity level, which can be unclassified, confidential, secret, or top secret.

The MAC model places various restrictions on user actions that prevent dynamic manipulation of the underlying permissions. This requires large parts of the OS and associated utilities to be "trusted" while assigning and enforcing secure levels by the system. Trusted components are usually a form of database and processes, such as releasing cryptographic processes that are placed outside of the MAC model due to their violation of MAC principles. The MAC model is simple but highly secure. It ensures the integrity of information by preventing unauthorized users from making changes to the information. Thus, it prevents the flow of information between users. However, the main disadvantage of the MAC model is that it does not ensure fine-grained access control as well as duty separation. Also, the MAC systems are usually high priced and difficult to use due to their reliance on the trusted components and their needs of applications for the MAC labels and properties.

3.4 | Role-based access control

The RBAC model aims to limit the users' access to sensitive information within an enterprise. In this model, the accesses to data and resources are based on predefined roles assigned to users by the system administrator. The RBAC model is a type of access control with which employees of an organization have different roles. These roles are specified based on the duties assigned to employees. Thus, access to resources is attributed to roles and not people directly. If the user changes his work in the enterprise, his role and permissions are changed accordingly. With this model, users cannot pass their permissions to other users. This represents the difference between RBAC and DAC.

The main components of the RBAC model are the following: user, role, permission, operation, and object. The RBAC model provides a mean for the relationships between these components, by illustrating the relationship between the employees of an organization and the rights attributed to them, as shown in Figure 4.

Several commercial systems have used RBAC as a security model. Baldwin[20] proposed a security policy for a database system using RBAC. Ramaswamy and Sandhu[21] studied the implementation of RBAC features in three popular databases: Informix Online Dynamic Server Version, Oracle Enterprise Server Version, and Sybase Adaptive Server Release. The feature was categorized into three categories: user role assignment, support for role relationships and constraints, and assignable privileges.

RBAC is a flexible model. When a person joins the organization, the system administrator assigns him a role based on his work. If the job of this employee changes, the system administrator assigns him a new role. Several employees with different roles in an organization can share common operations. It is unnecessary and repetitive to define these common operations in all roles. This is how the concept "role hierarchy" of the RBAC model comes from. With this concept, a parent role can implicitly contain other roles. The parent role will contain all the operations, objects, and permissions of its descendant roles. For instance, in a university, the role of instructor includes the following roles: an assistant professor, an associate professor, and a full professor, as shown in Figure 5.

RBAC model evolution was done under four main models. RBAC0 was the initial model. It consists of separating the duties and the least privileges. RBAC0 is not a hierarchical model; hence, users were assigned direct permissions. RBAC1 introduced the use of hierarchies based on the responsibilities and job levels inside organizations.[22] RBAC2 introduced the concept of constraints acting as limiters to enforce policies and regulate resources access based on certain criteria. Finally, RBAC3 covered all of the components in the previous RBAC models. Therefore, allowing a full hierarchical structure. Bertino et al[23] proposed the temporal RBAC model that extends the RBAC model and enables and disables a role for a user while considering the temporal dimension.

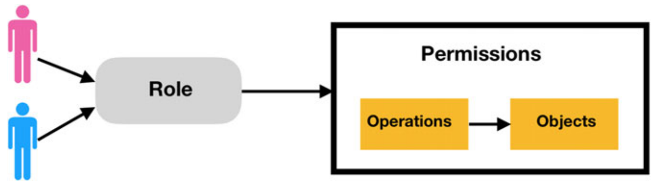

FIGURE 4 Role-based access control model

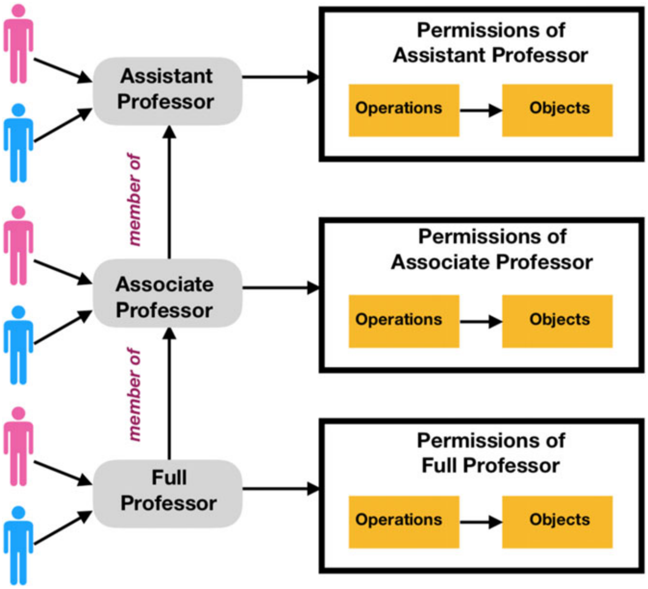

FIGURE 5 Example of RBAC2 hierarchical model

Based on the RBAC model, Premarathne et al[24] presented a hybrid cryptography access control model for cloud-based electronic health record (EHR) systems. In the proposed model, user authentication is based on biometrics and localization. Almutairi et al[25] proposed a distributed access control architecture for cloud computing based on the RBAC model. The goal of the proposed architecture is to meet the cloud access control requirements, such as multitenancy and virtualization, decentralized administration, secure distributed collaboration, credential federation, and constraint specification.

The advantages of RBAC models are multiples. They can be summarized as follows:

- Ease of implementation: The RBAC model needs to be deployed through roles engineering, to reflect all positions inside the organizational policy. This task requires a lot of research and testing to ensure that the concept of least privileged is achieved through role design. Once the role is tested and implemented, administrators can benefit from the new design, which allows less human intervention when updating access requests. Moving the user from current role to another is hence an easy task and it is done through the disassociation from the user's old role and association of a new role permission to access through the assigned role. Moreover, the disassociation of the user from his current role makes him unable to access the system and benefit from the assigned permissions. Should the user leave the work, the task of disassociation of the user from the role is easy and makes the access impossible even if their account accidentally remains active.
- Hierarchy and rights inheritance: RBAC3 supports a hierarchal framework, which can be used to ease association by allowing permissions to spill down to subsidiary objects. In Novell Netware, an example would be the rights coming out of an authorization unit down into the users arranged underneath it. The other advantage of this comes into place as for role design; this dynamic framework can immensely diminish the number of roles made. Another advantage is that different roles can be associated with each other to allow greater functionality for the end-user.
- Separation of duties: RBAC3 permits and authorizes separation of duties through constraints, which implies that a user with a specific employment job role cannot be in another role at the same time. This feature is useful and required, particularly, in health frameworks.
- Scalability: RBAC3 is scalable. It allows well definition and documentation of policies within enterprises. In an enterprise, roles can be made by engineers and changed similarly as required. One advantage is to give users a very low level of administration. Hence, individual account administration is reduced or even eliminated. As the organization grows, more roles may be required.
- Security: Role planning before implementation leads to some security vulnerabilities inherited from DAC, for example, administrator permission errors during logging in and out.

Figure 6 shows the RBAC model workflow, where both RBAC decision and historic behavior are triggered.

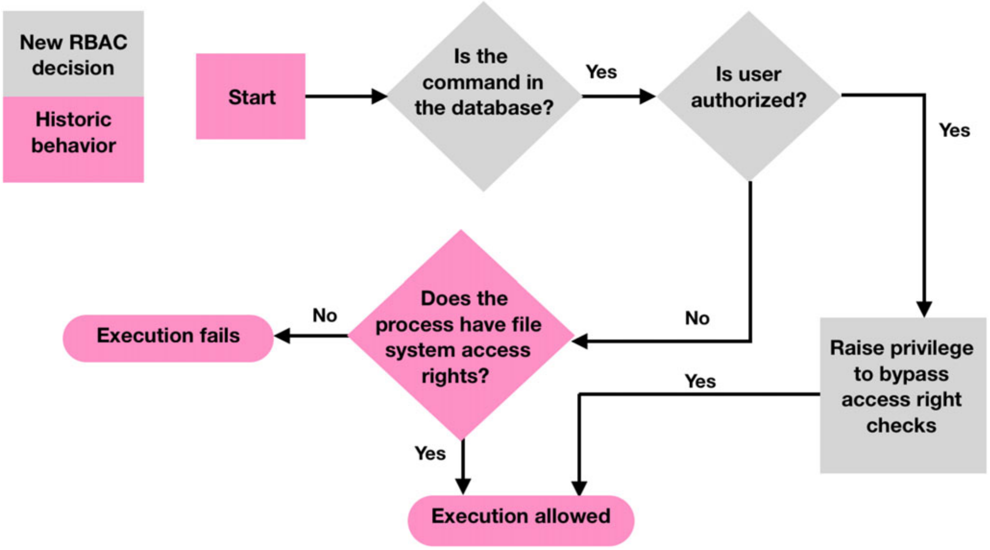
New RBAC decision
Historic behavior
Start
Is the command in the database?
Yes
Is user authorized?
Yes
Raise privilege to bypass access right checks
Yes
No
Does the process have file system access rights?
No
Execution fails
Yes
Execution allowed

FIGURE 6 Role-based access control (RBAC) work flow and decision-making

The disadvantages of the RBAC model are as follows:

- RBAC is static. It cannot use contextual information such as time, user location, and device type which are used to restrict the access to resources and reduce the probability of information leakage.
- RBAC model does not provide the delegation capability.
- The permissions associated with each role can be deleted or altered based on the role privilege.
- RBAC is coarse-grained and static and ignores resources metadata. It is hard to maintain and has no dynamic flow when dealing with multiple computer systems.
- RBAC, as well as the DAC model presented previously, are error prone. In large organizations where the number of employees is large and dynamic, the management of RBAC and DAC models becomes difficult, errors prone and hardly scalable. In fact, every time an employee leaves the organization, or changes his role, or even be assigned with a newly added role, the list of users privileges has to be updated. In this case, errors may occur due to human mistakes during data entry.

3.5 | Attribute-based access control

The ABAC model grants or denies a user request to access a specific resource based on the attributes associated with the user and the resource. ABAC model allows the combination of different parameters and rules to express policies.[26] It incorporates more attributes such as the location, authentication level, qualifications, time, etc. ABAC components are the users, subjects, objects, user attributes, subject attributes, object attributes, permissions, authorization policies, and constraint checking policies. In the ABAC model, every user is associated with some attributes related to its properties, such as its role, group, department, project, etc. Subjects are made by users to perform out a few activities in the framework. Alshiky et al[27] implemented an access control mechanism based on the ABAC scheme to protect the EHR data in fog computing environment. ABAC model is implemented in a Fog device representing the edge of the network. If the request's attributes meet the predefined policies scheme, the fog node will grant permission to the user.

The main advantage of ABAC is ensuring that the right information is only accessible by the right people and only when they need it. Multiple other advantages of the ABAC model were mentioned by NIST. One can cite the following[26]:

- Single point provisioning of users: The ABAC system administrator is not obliged to check users' account, to assign roles, or to modify their access control list based on approval processes. The ABAC system is able to know what is accessible to the user based on the policies assigned to the application.
- Dynamic access control: Access control is dynamically made based on the most updated policies. Digital policies always change to address security alarms, which include conditions such as the national level of security. The ABAC model uses these updates as input data for policy decisions, which allows flexible control depending on the organization change.

- Finer-grained access control: RBAC may results in "roles explosion" when federal agencies administrators create roles for a small group of people despite the many updates on the access levels. ABAC allows accurate access control by extracting from a higher set of attributes to take a decision, generating a bigger set of probable rules and choices without managing groups and roles.

There is a considerable list of problems related to ABAC systems and applications.

- Untraditional way of users' authorization: Unlike traditional IAM access control mechanisms, permissions and roles assignments in the ABAC model are handled by the security team. Access reviews are performed on a regular basis where user-role assignments are checked and approved by managers. However, in ABAC, the user-role assignment is directly allocated through roles and permissions. Users' entitlements are the result of a runtime authorization request evaluated against a set of policies. This new form of assignment makes access reviews, provisioning, and deprovisioning insufficient. In this case, ABAC requires a new process for the above actions; therefore, new authorization requirements should be implemented.
- Lack of requirements: In the traditional access control mechanisms, requirements are handled by applications developers who implement the requirements as codes inside the applications. In ABAC, authorization requirements are gathered and coded as authorization policies centrally managed. Therefore, new steps should be implemented: use case definition and authorization requirements gathering.
- Complex ownership of authorization: Most of the ownership and responsibility in traditional IAM lies in the central IAM team. This is done by defining coarse-grained access with the RBAC system and then allowing the developers to implement fine-grained controls in the applications. In the ABAC model, the entire authorization logic is expressed inside the authorization policies. In other words, the central IT team, application owners, and business analysts should work together to define the requirements and to agree on the ownership.
- Emulating and representing traditional models: The ability of ABAC model to emulate traditional access control mechanisms models makes it the more general access control model. However, there is no real proof to support this claim. In this light,[28] proved in their work how ABAC Alfa model can be constrained to model DAC, MAC and hierarchical RBAC models.
- Hierarchical ABAC: In a hierarchical RBAC, roles are related in a way similar to that of real organizations, which simplifies the administration on both engineering and reviewability of existing role-based policies levels. The majority of "pure" ABAC models are missing this kind of inheritance and expressiveness. While a role can be easily modeled as a single attribute of a subject, this simplistic representation cannot emulate RBAC's hierarchical nature without allowing for complex data types in the value of an attribute [28] or unmaintainable complex policies. Pure ABAC needs simpler ways in order to provide hierarchical administration and to be able to compete with RBAC models. "Attribute users groups" which are hierarchical groups that inherit sets of attributes from their parent groups and allocate them to their members, may provide a solution. This technique could also work for objects and other access control entities into which the attributes may be assigned. Another solution is to allow the attributes to have direct inheritance relationships with other attributes, such that a child attribute supersedes the parent attribute in policies. However, this leaves the attributes with no value and limits ABAC's usefulness.
- Auditability: The ability to determine the users who can access a set of resources is a major aspect of access control for legal and security reasons. This feature is easily insured in the RBAC model. However, in the ABAC model, ensuring this feature is more complicated. ABAC being an identity-less access control system, even when all users the identities and their assigned attributes are known, computing the resulting set of permissions for a given user is not trivial since all objects have to be checked against all relevant policies.
- Duties separation: To limit potential errors and fraud, many people can accomplish a specific and sensitive task at the same time. In the RBAC model, people are not allowed to be given conflicting roles that are provided by static duty separation. In the ABAC model, applying this concept is yet to be explored. Applying duty separation to ABAC is still an issue. In an attempt to solve this problem, Alipour and Sabbari[29] introduced "can't-perform" rules that prevent the subject from doing certain actions on specified resources. This solution requires knowing the subject and the possible conflicts of interest beforehand.
- Delegation: This feature allows a user to instantly designate his permissions and privileges to a more junior user. This is regularly proficient by empowering delegation of allotted roles under certain predefined limitations and renouncement conditions.[30-33] While the delegation is considered as ABAC-based encryption,[34,35] several attempts have been made to add the delegation capability to ABAC model.

- Scalability: Traditional access control mechanisms such as the RBAC model have been highly adopted to complex systems; however, the ABAC model is still not proven in terms of efficient scalability. ABAC model needs complex interconnections between access control entities that may be distributed on different network resources. In complex systems with hundreds of users, permissions, and policies, it is not clear how ABAC solutions can be managed in terms of the required administration and computing resources. Complex studies of large systems utilizing ABAC concepts are needed to calculate the feasibility and usability of ABAC.[26]

ABAC models are of two categories: General architecture and domain architecture. Tables 1, 2 and 3 present and compare several well-known ABAC architectures based on the following criteria: (1)objects attributes, (2) environment attributes, (3) user attributes, (4) functional specification, (5) connection attributes, (6) mutable attributes, (7) hierarchical, (8) policy language, (9) recursive rules, (10) user and object groups, (11) separation of duties, (12) delegation, (13) trust, (14) formal model, (15) administration model, and (16) complete model. The comparison done shows that the almost ABAC models had the object and user attributes, as well as they are considered a formal and complete model. However, none of these models has connection attributes, recursive rules, user and object groups, delegation, and trust.

The policy-based access control (PBAC) mechanism is a variant of the ABAC model. It is designed to help companies in implementing solid access controls based on a clear and well-defined policy and requirements. PBAC is considered a harmonized and standardized form of ABAC model at an enterprise level. PBAC gathers attributes from resources,

TABLE 1 ABAC model comparison - General architecture

	Authors					
Criterion	28	36	37	38	39	40
Objects attributes	✓	✗	✓	✓	✓	✓
Environment attributes	✗	✗	✗	✗	✓	✓
User attributes	✓	✓	✓	✓	✓	✓
Functional specification	✓	✗	✗	✗	✗	✗
Connection attributes	✗	✗	✗	✗	✗	✓
Mutable attributes	✗	✗	✗	✗	✗	✗
Hierarchical	✗	✗	✗	✗	✗	✗
Policy language	XACML	Undefined	N/A	Undefined	N/A	XACML
Recursive rule	✗	✓	✗	✗	✗	✗
User and object groups	✗	✗	✗	✗	✗	✓
Separation of duties	✗	✗	✗	✓	✗	✗
Delegation	✗	✗	✗	✗	✗	✗
Trust	✗	✗	✗	✗	✗	✗
Formal model	✓	✓	✓	✓	✗	✓
Administration model	✓	✗	N/A	✓	N/A	✓
Complete model	✓	✗	✓	✓	✓	✓

TABLE 2 Attribute-based access control model comparison: domain architecture (2005-2010)

	Authors			
Criterion	41	42-44	45	46
Object attributes	✓	✓	✗	✓
User attributes	✓	✓	✓	✓
Environment attributes	✓	✓	✗	✗
Connection attributes	✗	✗	✗	✗
Mutable attributes	✗	✗	✗	✗
Policy language	XACML	XACML	XACML	XACML
Hierarchical	✗	✗	✗	✗
Recursive rule	✗	✗	✗	✗
Trust	✗	✗	✗	✗
User and object groups	✗	✗	✗	✗
Separation of duties	✗	✗	✗	✗
Delegation	✗	✗	✗	✗
Functional specification	✗	✗	✗	✗
Formal model	✓	✓	✗	✓
Emulates traditional models	N/A	N/A	N/A	N/A
Administration model	✗	✗	✗	✗
Complete model	✗	✓	✗	✗

TABLE 3 Attribute-based access control model comparison - Domain architecture (2010-2014)

	Authors					
Criterion	43	47	48	49	50	51
Object attributes	✓	✓	✓	✓	✓	✓
User attributes	✓	✓	✓	✓	✓	✓
Environment attributes	✓	✓	✓	✓	✗	✓
Connection attributes	✗	✗	✗	✗	✗	✗
Mutable attributes	✗	✗	✗	✗	✓	✗
Policy language	XACML	Class Algebra	XACML	N/A	Undefined	XACML
Hierarchical	✗	✗	✗	✗	✗	✗
Recursive rule	✗	✗	✗	✗	✗	✗
Trust	✗	✗	✗	✗	✓	✗
User and object groups	✗	✗	✗	✗	✗	✗
Separation of duties	✗	✗	✗	✗	✗	✗
Delegation	✗	✗	✗	✗	✗	✗
Functional specification	✗	✗	✗	✗	✗	✗
Formal model	✗	✗	✓	✗	✓	✗
Emulates traditional models	N/A	N/A	N/A	N/A	N/A	N/A
Administration model	✗	✗	✗	✗	✗	✗
Complete model	✗	✗	✗	✗	✗	✗

environments, and requesters with specific information on the conditions under which the access request was made. PBAC also utilizes rule sets that tell whether, under an organizational policy, the access is allowed for those attributes under those conditions.[14]

Under the PBAC model, enterprises might have only one policy that manages access to sensitive or critical resources regardless of the location or the owner of the data. PBAC is more complicated than ABAC; PBAC attributes should be designed, deployed, and maintained in enterprise high-level systems. Examples would be like databases, directory services, and other middleware and management applications, all of which must be integrated. Moreover, PBAC requires a complicated algorithm to manage access based on attributes. It also requires a mechanism to build and to manage policy rules in an unambiguous way; otherwise, illegal access to information resources can be achieved. Policy creation is not easy even with the use of ACML. Attributes used across the enterprise must be the same and from authoritative sources only.

3.6 | Attribute-based encryption

There is a huge amount of sensitive data stored by third parties on the web such as emails stored on Google, MSN, etc. The security of this data will be therefore in question. One of the solutions to solve this problem and protect the data from any unauthorized use and/or loss is to store the data in encrypted form. The main drawback of this solution is that it does not allow sharing the encrypted data at a fine-grained level. For instance, consider a database of encrypted medical records. The records are labeled with the date, time, patient sex, patient age, etc, and the database is encrypted using a traditional public key encryption scheme. Now suppose that an analyst asks the owner of these data to explore and exploit the records of elderly patients (having age >=75). In this case, only two options are possible for the data owner. First is to give the analyst the key and thus allow him to decrypt all the medical records in the database. In this case, the analyst could access all the records and not only to those of elderly patients. Second is to not give the analyst the private key and therefore to deny his request. To solve this problem, Sahai and Waters[52] introduced the attribute-based encryption (ABE) that encrypts and decrypts the data using a set of user attributes. Actually, the user's keys and ciphertexts are both labeled with sets of attributes. These attributes represent the user identity, and they are selected by the encryptor. A particular key can decrypt a particular ciphertext if there is a match between the attributes of the ciphertext and those of the user's private key. Returning to the previous example, let us assume that the database was encrypted using the ABE model. Thus, each medical record is associated with a set of attributes (date, time, patient sex, patient age, etc). When the analyst asks the data owner to access a specific portion of these data, ie, those of elderly patients, the data owner could create a private key that can decrypt only the ciphertexts having the following attribute: Age $> = 75$. Zhu et al[53] presented a temporal access control model for cloud computing that associates an access policy on the attributes to each data item stored in the cloud. The data attributes are of the temporal type, ie, period-of-validity, opening hours, and service hours. The proposed system combines three advanced cryptographic techniques: integer comparison, current-time reencryption, and ABE. Yan et al[54] presented a heterogeneous multidimensional access control scheme to flexibly control data access based on trust and reputation in cloud computing based on the policies and strategies set by the data owner. In this system, the data owner encrypts its data with a secret key that can be divided into multiple parts so that to support various access control policies.

There are five variants of the ABE model: simple ABE, key-policy ABE (KP-ABE), ciphertext-policy ABE (CP-ABE), ABE with nonmonotonic access, and hierarchical ABE (HABE). Lee et al[55] presented a survey and compared these five ABE models based on several criteria. Goyal et al[56] proposed a variation of the classical ABE model, the so-called KP-ABE model, which is designed for one-to-many communications. In this system, each ciphertext is labeled with a set of attributes chosen by the encryptor, and each private key is associated with an access structure that specifies which type of ciphertexts it can decrypt. If the encrypted data attributes satisfy the access control policy built into the user's private key, then, the user can decrypt and access the data. Several models were proposed in the literature based on the KP-ABE model. One can cite the work of Yu et al[57] where the data owner encrypts his data and shares them with other users by distributing keys to them. The keys contain the attribute-based access privileges.

Another variation of the ABE model, called CP-ABE, was presented by Bethencourt et al[58] Unlike the KP-ABE model, the access control policy in the CP-ABE model is built into the encrypted data while the set of attributes is specified in the user's private key. If the attributes satisfy the access policy of the encrypted data, then the user can decrypt and access the encrypted data. Yang et al[59] proposed a time-domain attribute-based access control model based on the CP-ABE scheme. The goal is to ensure the security of shared video in cloud-based multimedia systems where some multimedia contents may be time sensitive and have to be accessed by the authorized user only during a particular time period. A multiauthority access control scheme in fog-cloud computing based on the CP-ABE model, the so-called VO-MAACS, was implemented by Fan et al.[60] In this scheme, the encryption and decryption of the data are performed in fog devices. At first, the data owner defines the access control policy. Second, he uses it to encrypt the data before uploading them to the cloud. When a user wants to access these data, he will have to use two keys: a proxy key and a secret key. To download the encrypted data from the cloud, the user asks the fog device to decrypt the data with its proxy key. Then, the fog device checks if the user attributes satisfy the access policy defined by the data owner. If that is the case, the fog device decrypts the ciphertext and sends the partially decrypted data to the user. This later will use finally its secret key to recover the data. Bobba et al[61] expanded the CP-ABE scheme and modeled the ciphertext-policy attribute-set–based encryption (CP-ASBE) to handle attribute revocations in a system. CP-ASBE organizes the user attributes into a recursive set structure, so that a particular attribute may contain several values, making this model quite flexible. Several other access control models derivated from the CP-ABE scheme have been proposed in the literature.[34,62-73]

To achieve a more fine-grained access control mechanism in the cloud storage services and to protect the privacy and security of the data stored in the cloud, Wang et al[74] proposed to combine the hierarchical identity-based encryption[75] and CP-ABE, the so-called HABE. Wan et al[76] extended the work of Bobba et al[61] and designed the hierarchical attribute-set-based encryption for more flexible and scalable access control in cloud computing. Jung et al[77] proposed an anonymous control system that addresses both the data privacy and the user in the cloud storage server. Jung et al described and demonstrated the security and feasibility of the scheme. Liu et al[78] proposed a multifactor authentication system for web-based cloud computing service. In such a system, more than one authentication technique is implemented and used into the cloud device. Ostrovsky et al[79] proposed a nonmonotonic access scheme based on the ABE system, which includes nonmonotonic access structures.

In the following, we compare the five ABE variants models according to the following criteria:

- C1 - Data confidentiality: This feature ensures that once the users' data are encrypted and uploaded into the cloud, they cannot be accessed by unauthorized parties as well as by the CSP.
- C2 - Fine-grained access control: The access control model should be fine-grained to become a user-specific rule. Actually, in an organization, and in the same group of users, different users may have different access permissions according to their roles.
- C3 - User accountability: It represents the ability of the access control system to trace the users' activities. The accountability is preserved due to the auditing.
- C4 - User revocation: If the user leaves the organization, the access control system must be able to directly withdraw its permissions and access rights.
- C5 - Scalability: Due to the dynamic business needs of the organizations, the number of users and their assigned roles could change frequently; thus, the performance of the access control model should not be affected.

Table 4 shows this comparison.

TABLE 4 Comparison of attribution-based encryption (ABE) models

	Criterion				
ABE model	**C1**	**C2**	**C3**	**C4**	**C5**
Simple ABE	✓	✓	✗	✗	✗
ABE with nonmonotonic access	✓	✓	✗	✓	✗
KP-ABE	✓	✓	✗	✓	✗
CP-ABE	✓	✓	✓	✓	✗
HABE	✓	✓	✓	✓	✓

3.7 | Federated identity management

Nowadays, with the diversity of sites and applications being used, global digital identities are spread over the web. Credentials should be created for each user visiting a new web page. These credentials are stored on that website. Each time end users visit this website, they have to reenter these credentials, and this authentication should be done as well for each new accessed site, even in the case where these different accessed sites are managed by the same organization. This makes this authentication process impractical, and therefore, requires the implementation of a new authentication system. Realizing this problem, researchers start to develop new authentication systems. That made the concept of federated identity management (FIM) very difficult to apply. The same origin policy forbids the access of the information stored on the end user's computer by another user except for the original creator of that data. This principle states that each domain is independent of the others. Thus, a specific domain cannot access customer credentials and transfer them to another domain. If so, the whole concept of Internet security would be questionable. As a result, companies with multiple domains that want an easy transmission between its domains have to find a secure solution permitting this transfer of information. Here, we explain the FIM systems that allow secure transfer of users' credentials data without violating the original policy.[80]

When the user tries to log into an application or a website, it sends a request for authentication to the authorization server. Based on the decision of the authorization server, the user will be permitted to access the application or not. If two or several domains are related and managed by the same authorization server, the user can log in into one of these domains and access the other domains without the need to authenticate for a second time. That is what we call the "single sign-on" (SSO). There are many SSO providers that provide this kind of service to the webmasters. Some SSO idPs are based on enterprise-focused systems, while others use decentralized systems. Nowadays, systems do not have passwords. Clients would no longere need a set of credentials to access different applications. The possession of one tool such as cell phones or unique fingerprints enables the users to move across different domains. The choice of the idP by the clients depends on the benefits and the limitations of the selected provider.

As a definition, the federated identity is a system that takes place between the organization and all its running applications. It certifies the authenticity of users by confirming the username and the related password they have entered. By this function, we can refer to the federated idP as a middleware. The users can access their application by using their existing active directory credentials through the federated identity. As a result, the users' authentication is done via on-premises active directory services.

Several benefits of the FIM model are identified:

- SSO: With the diversity of devices and applications, the users are required to create and hold numerous login credentials. On the one hand, it is difficult from a user perspective to remember all these credentials. On the other hand, it is a waste of time for IT administrators to control all these accesses since they have to manage multiple users' identities across different applications. The federated identity also called "SSO" can be considered a solution. This latter could be implemented using existing active directory credentials. This model allows a true SSO. The users can have the same password for all cloud applications and other third-party cloud applications. This makes the IT user experience more convenient, simpler, and quicker.
- Reduced security risks: Federated identity increases the security level. By identifying the authentication process within on-premises active directory, IT administrators do not have to synchronize different passwords existing on the cloud active directory. Actually, the authentication policy is stored on-premises, behind the firewall. The use of an SSO model presents a win-win position for both users and IT administrators. In fact, creating multiple login credentials expose the organization to serious risks and increases the potential use of weak passwords by the users. The use of an SSO policy is more convenient for both employees and IT teams and helps to create a strong security policy.
- Increased organizational productivity: The use of cloud-based applications helps organizations to increase their productivity. Actually, if IT teams have to deal constantly with multiple applications logins, this will increase the

administrative tasks within the organizations. The login process can be simplified by using the federated SSO policy. As a result, the company's productivity will be improved and the user will have only to remember his “domain credentials.” Ultimately, FIM is cheaper and more secure since it does not need to manage individual cloud-based accounts.

The FIM architecture consists of the following entities:

- Users: Each user is represented by a set of attributes that describe the qualities (ie, age), the circumstances (ie, the employer), the behaviors (ie, shopping) or the assigned values (ie, USERID) of the user. The number of attributes associated with identity is not restricted, and a single user can have several identities at the same time. Identity management systems allow users to choose among multiple digital identities.
- SPs: They give the authorization to the users based on authentication assertions. Each SP has its own identity management, and the users have separate identities for each SP they are dealing with.
- IdPs: They can be autonomous parties or the SPs themselves. IdPs aim to authenticate the users and to store and manage the collections of attributes associated with these users. IdPs have the ability to configure all the users' identities and thus to create, update, release, and delete any record whether it was attribute or identity.
- Trust establishment in FIM: Two methods of trust establishment are identified: static trust establishment and dynamic trust establishment. With the static trust establishment, the trust is predefined between IdPs and SPs. This trust can be through negotiation between the two parties or during the implementation phase. Many models can be used to implement the static trust establishment. Chen et al[81] proposed a model that allows the interoperation paths to be discovered inside IdPs based on different circles of trust (CoT). The model describes how trust can be established between CoTs to allow path interoperation and discovery. Authentication assurance level (AAL) conversion is designed and role mapping is also implemented to improve the level of interoperation security. Jiang et al[82] implemented a new entity called trust service provider (TSP) which allows, at runtime, to establish and manage the trust relationship between federated entities. The TSP requires registration to obtain a certificate; hence, the parties can communicate through a secure and private channel. TSP is considered the third trust party where federated parties share their metadata.

  In case there is a large number of IdPs and SPs, the static trust establishment model would not be the right choice. This gives rise to the concept of dynamic trust establishment. This model is based on the metadata provided by IdPs and SPs along with their service-level agreement (SLA) and reputation. There are several models of dynamic trust. Bhonsle et al[83] implemented the Efficient Trust and Identity Management System (ETIS) where the trust third party is not mandatory. ETIS allows SPs to establish trust between themselves without going through the third party. Mármol et al[84] suggested a model called Trust and Reputation Model for Identity Management Systems (TRIMS) that offers an acceptable security level where multiple domains can decide about their reliability and exchange sensitive user attributes. When a client requests web services from a web service provider (WSP), it requests by his turn some information from the IdP. In this scenario, IdPs acts as a basic role to hold identity information based on the users' requests. Kanwal et al[85] proposed a trust establishment model that evaluates the trust level of CSP. The model has the following submodules: registration management module, SLA management module, feedback management module, and trust management module. However, this model does not monitor or update the trust score of CSP.

The comparison of FIM models is based on several factors, as follows:

- C1 - Trust management: In every structure, there is an object responsible to achieve the communication or trust creation between IdPs and SPs. This could be done by a centralized unit, or between IdPs and SPs themselves (peer to peer (P2P)). User requirements allow us to choose which form to use. This choice is also influenced by the number of IdPs and SPs. Many solutions have proposed centralized forms for the organization of IdPs and SPs, but these solutions might not be feasible in a large network of IdPs and SPs, as the central unit might have to tolerate a lot of data processing load, causing an incompetent structure. However, if all the IdPs and SPs interconnect straightly (P2P), the resolution becomes more scalable, but the trust establishment may be difficult to achieve.
- C2 - Trust establishment: In FIM, the trust is established offline through some trust cooperation procedure. IdPs and SPs might meet to work on a deal and sign an agreement for trust establishment. Sometimes, IdPs or SPs have to register with the centralized unit so that other entities could trust it, but it is impossible to have one centralized unit to serve all IdPs and SPs at the same time. The number of parties working in federation might be smaller than the number of IdPs and SPs combined together. In this case, the user might use static trust established to provide more confidence and legitimate sense towards the SPs.
- C3 - User privacy: This can be a major concern when it comes to malicious SPs. The worst case scenario can be an identity theft of users, password stealing, fraud activities, and money laundry financial transactions.

TABLE 5 Comparison of trust-based federated identity management (FIM) models

	Criterion					
FIM model	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
[81]	Peer-to-Peer	Static	✗	✓	✗	✗
[82]	Centralized	Static	✗	✗	✗	✗
[83]	Peer-to-Peer	Dynamic	✓	✗	✗	✗
[84]	Centralized	Dynamic	✓	✗	✓	✗
[85]	Centralized	Dynamic	✓	✗	✓	✗
[86]	Centralized	Dynamic	✗	✗	✗	✗
[87]	Centralized	Static	✓	✗	✗	✗
[88]	N/A	Dynamic/Static	✓	✗	✗	✗
[89]	Centralized	Static	✓	✗	✗	✗
[90]	Centralized	Dynamic	✗	✗	✗	✗
[91]	Centralized	Dynamic	✗	✗	✗	✗

- C4 - Reliable access rights across CoTs: Users might be assigned roles and privileges in the CoT where they belong to. However, when the users are allowed to get services of an SP inside or outside the CoT, it will become unstable in terms of the number and level of rights assigned to the users. This scenario may lead to some sort of security attacks known as escalation of privileges attack that may lead to security compromise inside the system.
- C5 - Continuous trust monitoring: Runtime trust monitoring can be done through multiple frameworks in order to keep evaluating the metrics and getting results of the trust relationship. Quality of services provided by SPs might affect as well the trust, which may lead to the degradation in the trust relationship.
- C6 - Adaptation to unexpected changes: The entities should work in dynamic environments where a lot of changes might occur without previous notifications. Therefore, FIM systems should be adaptable to any potential changes or unwanted situations. Situations can be geolocation problems or anything related to information system degradation.

Table 5 shows the FIM models comparison based on the factors described above.

3.8 | Evaluation of access control mechanisms

Based on a literature review, we identified the following set of fundamental requirements for an access control mechanism in cloud computing systems:

- Least privilege principle: The user should have access to only the required information and resources to perform a specific task. Therefore, the user is granted the only needed permissions to accomplish its task, even if more advanced permissions are associated with it.
- Separation of duties: It represents the ability of the access control model to prohibit unauthorized users from accessing the requested resources. Thus, only the permitted users that are duty-related to the resources will be granted the permissions to access the related resources.
- Capability delegation: The capability is the ability of a user to access an object (file or resources) in a system. Capability delegation is the ability of a user to delegate its capability to other users. In addition to the delegation, there are other capability operations. The capability revocation allows the user to revoke the features that it has delegated previously to other users. Temporary deactivation of the delegated capabilities is achieved by the disable capability. Finally, the delete capability deactivates the delegated capabilities permanently.
- Auditing: This criterion is very important for securing cloud systems. It allows the access control system to monitor its state by recording the denied and granted user access requests.
- Policy management: It depicts the ability of the access control model to uses different security strategies having different rules while preventing the conflicts between them.
- Scalability: In cloud computing environments, due to the dynamic business needs of the organizations, the number of users and their assigned roles could change frequently. Therefore, an access control model for cloud systems is assumed to be scalable in terms of the number of users.
- Authentication feature: It represents the ability of the access control mechanism to support identification and authentication functions. Actually, cloud computing systems need strong authentication mechanisms to manage users identities and authenticating them.
- OS compatibility: It depicts the ability of the access control mechanism used by the cloud computing system to work with a variety of OSs.

TABLE 6 Comparison of access control mechanisms (ACMs)

	ACM					
Criterion	**MAC**	**DAC**	**RBAC**	**ABAC**	**ABE**	**FIM**
Least privilege principle	✗	✗	✓	✓	✓	✓
Separation of duties	✗	✗	✓	✓	✓	✓
Capability delegation	✗	✓	✗	✗	✗	✗
Auditing	✓	✓	✓	✓	✓	✓
Policy management	✗	✗	✗	✗	✗	✓
Scalability	✗	✗	✓	✗	✗	✓
Authentication feature	✗	✗	✗	✗	✓	✓
OS compatibility	✓	✗	✓	✗	✗	✗
Bypass	N/A	N/A	N/A	N/A	N/A	N/A
Safety	✓	✗	✓	✓	✓	✗
Flexibility of configuration	✗	✓	✓	✗	✗	✓
Response time	N/A	N/A	N/A	N/A	N/A	N/A

Abbreviations: ABAC, attribute-based access control; ABE, attribute-based access encryption; DAC, discretionary access control; FIM, federated identity management; MAC, mandatory access control; OS, operating system; RBAC, role-based access control.

- Bypass: It is the capacity of the access control system to bypass the policy rules during critical situations while proportioning with the tolerance risk of the enterprise.
- Safety: An access control system is considered as safe if its configuration prevents the leakage of permissions to unauthorized users.
- Flexibility of configuration: This feature allows the access control model to deal with dynamic environments such as cloud computing systems, and allows it to be flexible in configuration, as well as easy in both installation and uninstallation.
- Response time: In cloud computing systems, there is a significant amount of users. Once authenticated, the access control model in the cloud system has to accept or deny their requests. The response time criterion describes the ability of the access control model to answer the users' requests within a reasonable time that meets the needs of the enterprise.

In Table 6, we compare, based on a literature review, the access control mechanisms presented previously against the criteria introduced above.

When it comes to choosing the right access control mechanism that fits a cloud system, several decision factors will be involved, such as the business nature, security procedures within the cloud, the number of users, etc. The following steps describe the selection procedure of the appropriate access control system:

- Needs and risks analysis: The first step when selecting an access control system is to evaluate the needs and threats. What is the information to be protected? Is it sensitive?
- Requirements analysis: The next step is to define clearly the requirements for the access control system, based on Table 6. Several questions may arise: What are the expectations from the access control system? What are the troubles to be averted? How the new access control system should improve the performance and efficiency of the organization?
- Find the right supplier: Once the requirements are defined, we can seek the manufacturers. In this stage, some factors have to be taken into consideration, such as the complexity of the system design, its ease of installation, and the availability of system documentation.

4 | CONCLUSION AND FUTURE DIRECTIONS

Data integrity preservation intends to protect the data from unauthorized deletion, modification or fabrication. It is achieved using database constraints and transactions accomplished by a database management system. When the private data of the users are stored in the cloud, the confidentiality of these data becomes essential to increase the reliability of the cloud. Data confidentiality can be ensured with the use of access control policies. In this work, we presented an in-depth study of the access control mechanisms in cloud systems. Throughout this work, we have studied and analyzed the different existing classical models. We have also presented the essential criteria that an access control mechanism for cloud computing systems must satisfy, and finally, we have compared the discussed access control models against these criteria.

We summarize in Table 7 the advantages and disadvantages of the access control mechanisms presented in this paper.

TABLE 7 Advantages and disadvantages of access control mechanisms (ACMs)

ACM	Advantages	Disadvantages	Applications
MAC	Scalable Secure Full control by administrators only	Hard implementation Relies on system to control access Not flexible Supports limited number of users	Government organizations Military Critical missions
DAC	Easy implementation Highly flexible Protect users from unauthorized access	Prone to errors Relies on object owner Not scalable Susceptible to Trojan horse attacks Difficult in maintenance and verification	Web applications OS: Linux, Unix, Netware Critical missions
RBAC	Easy implementation Hierarchy and rights inheritance Duties separation Scalable Highly secure	Hard to manage and maintain Hard to implement fine grained access control Prone to errors	Health care systems Academic institutions Banking systems
ABAC	Single point provisioning of users Dynamic access control Fine-grained access control	Lack of requirements Complex authorization ownership Complex auditability Complex delegation Hard scalability	Government organizations Health care systems Airlines companies Insurance systems Telecommunications carriers
ABE	Fine-grained access control Highly secure	Complex delegation Hard scalability Not flexible	Security of personal health record Audit Log applications Cloud applications
FIM	Reduced security risks Delegated administration Increased organizational productivity	Trust management Continuous trust monitoring Trust estabilishment User privacy Adaptation to unexpected changes	Web applications Banking systems Cloud applications Mobile applications

Abbreviations: ABAC, attribute-based access control; ABE, attribute-based access encryption; DAC, discretionary access control; FIM, federated identity management; MAC, mandatory access control; OS, operating system; RBAC, role-based access control.

In cloud systems, access control is a very important step for protecting the security of users data. Thus, data storage and accessibility in the cloud must be well accomplished. At the base of what has been presented above, many research axes have been identified. They are summarized as follows:

- Supporting user revocation is a requirement for cloud systems, which represents an important challenge in existing ABE models. Therefore, implementing an efficient user revocation mechanism on the top of the ABE model is one of the future directions.
- There are actually few attempts to integrate RBAC and ABAC models. Incorporating these two models helps to combine their advantages while overcoming their limitations.[92] Thus, designing and developing a more scalable, flexible, fine-grained, and auditable access control mechanism in cloud systems by combining RBAC and ABAC models should be considered in the prospective studies.
- Cloud systems are dynamic environments since the users, resources, and services change frequently. Traditional access control models cannot satisfy the dynamic cloud security needs since they use static security policies,[93] thus the need for designing new dynamic access control schemes. Dynamic access control models allow using several security policies at the same time to grant or deny a user request to access a specific resource. Several dynamic characteristics of the user have to be thus considered and evaluated in real-time in order to take the decision.
- Given the difference of the services and security and compatibility issues between different CSPs, cloud computing standardization becomes important.[94] Actually, the difficulty of extracting and moving the data and the services from a cloud to another one is preventing some organizations from adopting cloud computing.[95] This risk should be considered when subscribing to the cloud services since there are no APIs for the data and applications in cloud computing, which limits their portability between different clouds providers. Therefore, if a company wants to change its cloud supplier or if this latter goes bankrupt, the transfer of the data of the company from the current cloud to another one will be a complex task and it will require a significant fee. Standardizing security policies and access control models in

cloud systems will help cloud users to evaluate the security of their cloud provider. Therefore, unified technical standards and industry specifications for cloud access control mechanisms including a reference access control architecture and security standards must be specified.[96] Further efforts must consider this issue.

- The fast development of mobile devices and mobile applications in recent years has led to the development of Mobile Cloud Computing (MCC). While MCC inherit all the capabilities of Cloud computing, but, it also takes all its security. Applying the access control mechanisms designed for cloud systems to MCC is not reliable for several reasons: (1) Mobile devices are limited in terms of resources, and therefore, access control mechanisms should not be time-consuming; (2) the security of data transfer between the mobile applications and the cloud should be considered; and (3) the security of MCC involves the security of the data stored in the cloud and the applications in the mobile device.[97,98]

ORCID

Rayane El Sibai https://orcid.org/0000-0001-9948-3601

REFERENCES

1. Adnan NAN, Ariffin S. Big data security in the web-based cloud storage system using 3D-AES block cipher cryptography algorithm. In: Proceedings of the 4th International Conference on Soft Computing in Data Science; 2018; Bangkok, Thailand.
2. Rawashdeh EF, Abuqaddom II, Hudaib AA. Trust models for services in cloud environment: a survey. In: Proceedings of the 9th International Conference on Information and Communication systems (ICICS); 2018; Irbid, Jordan.
3. Avizienis A, Laprie J-C, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans Dependable Secure Comput*. 2004;1(1):11-33.
4. Namasudra S, Roy P. Secure and efficient data access control in cloud computing environment: a survey. *Multiagent Grid Syst*. 2016;12(2):69-90.
5. Charanya R, Aramudhan M. Survey on access control issues in cloud computing. In: Proceedings of the International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS); 2016; Pudukkottai, India.
6. Zafar F, Khan A, Malik SUR, et al. A survey of cloud computing data integrity schemes: design challenges, taxonomy and future trends. *Comput Secur*. 2017;65:29-49.
7. Armbrust M, Fox A, Griffith R, et al. A view of cloud computing. *Commun ACM*. 2010;53(4):50-58.
8. Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing. *J Netw Comput Appl*. 2011;34(1):1-11.
9. Liu F, Tong J, Mao J, et al. NIST cloud computing reference architecture. *NIST Special Publ*. 2011;500(2011):1-28.
10. Nepal S, Ranjan R, Choo KKR. Trustworthy processing of healthcare big data in hybrid clouds. *IEEE Cloud Comput*. 2015;2(2):78-84.
11. Dillon T, Wu C, Chang E. Cloud computing: issues and challenges. In: Proceedings of the 24th International Conference on Advanced Information Networking and Applications; 2010; Perth, Australia.
12. Senyo PK, Addae E, Boateng R. Cloud computing research: a review of research themes, frameworks, methods and future research directions. *Int J Inf Manag*. 2018;38(1):128-139.
13. Shyshkina M. The hybrid service model of electronic resources access in the cloud-based learning environment. arXiv preprint arXiv:1807.09264. 2018.
14. Moghaddam FF, Wieder P, Yahyapour R. An effective user revocation for policy-based access control schema in clouds. In: Proceedings of the 6th International Conference on Cloud Networking (CLOUDNET); 2017; Prague, Czech Republic.
15. Suhendra V. A survey on access control deployment. In: Proceedings of the International Conference on Security Technology; 2011; Jeju Island, South Korea.
16. Lockhart H, Campbell B. Security assertion markup language (saml) v2. 0 technical overview. *OASIS Comm Draft*. 2008;2:94-106.
17. Hu L, Ying S, Jia X, Zhao K. Towards an approach of semantic access control for cloud computing. In: Proceedings of the International Conference on Cloud Computing; 2009; Beijing, China.
18. Zamite J, Domingos D, Silva MJ, Santos C. Group-based discretionary access control for epidemiological resources. *Procedia Technology*. 2013;9:1149-1158.
19. Bell DE, La Padula LJ. Secure computer system: Unified exposition and multics interpretation. Bedford, MA: MITRE Corp; 1976.
20. Baldwin RW. Naming and grouping privileges to simplify security management in large databases. In: Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy; 1990; Oakland, CA.
21. Ramaswamy C, Sandhu R. Role-based access control features in commercial database management systems. In: Proceedings of the 21st Nat'l Information Systems Security Conference; 1998; Arlington, VA.
22. Ferraiolo DF, Sandhu R, Gavrila S, Kuhn DR, Chandramouli R. Proposed NIST standard for role-based access control. *ACM Trans Inf Syst Secur*. 2001;4(3):224-274.
23. Bertino E, Bonatti PA, Ferrari E. TRBAC: a temporal role-based access control model. *ACM Trans Inf Syst Secur*. 2001;4(3):191-233.
24. Premarathne U, Abuadbba A, Alabdulatif A, et al. Hybrid cryptographic access control for cloud-based EHR systems. *IEEE Cloud Comput*. 2016;3(4):58-64.

25. Almutairi A, Sarfraz M, Basalamah S, Aref W, Ghafoor A. A distributed access control architecture for cloud computing. *IEEE Software*. 2012;29(2):36-44.
26. Hu VC, Kuhn DR, Ferraiolo DF, Voas J. Attribute-based access control. *Computer*. 2015;48(2):85-88.
27. Alshiky AM, Buhari SM, Barnawi A. Attribute-based access control (ABAC) for EHR in fog computing environment. *Int J Cloud Comput Serv Archit*. 2017;7(1):27-34.
28. Jin X, Krishnan R, Sandhu R. A unified attribute-based access control model covering DAC, MAC and RBAC. In: Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy; 2012; Paris, France.
29. Alipour HS, Sabbari M. Definition of action and attribute based access control rules for web services. In: Proceedings of the International Conference on Industrial Engineering and Operations Management; 2012; Istanbul, Turkey.
30. Barka E, Sandhu R. Framework for role-based delegation models. In: Proceedings of the 16th Annual Conference on Computer Security Applications (ACSAC); 2000; New Orleans, LA.
31. Wang H, Osborn S. Static and dynamic delegation in the role graph model. *IEEE Trans Knowl Data Eng*. 2011;23(10):1569-1582.
32. Zhang X, Oh S, Sandhu R. PBDM: a flexible delegation model in RBAC. In: Proceedings of the 8th ACM Symposium on Access Control Models and Technologies; 2003; Como, Italy.
33. Wang H, Osborn SL. Delegation in the role graph model. In: Proceedings of the 11th ACM Symposium on Access Control Models and Technologies; 2006; Lake Tahoe, CA.
34. Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Proceedings of the International Workshop on Public Key Cryptography; 2011; Taormina, Italy.
35. Servos D, Mohammed S, Fiaidhi J, Kim T. Extensions to ciphertext–policy attribute–based encryption to support distributed environments. *Int J Comput Appl Technol*. 2013;47(2-3):215-226.
36. Wang L, Wijesekera D, Jajodia S. A logic-based framework for attribute based access control. In: Proceedings of the 2004 ACM Workshop on Formal Methods in Security Engineering; 2004; Washington, DC.
37. Zhang X, Li Y, Nalla D. An attribute-based access matrix model. In: Proceedings of the ACM Symposium on Applied Computing; 2005; Santa Fe, NM.
38. Ferraiolo D, Atluri V, Gavrila S. The policy machine: a novel architecture and framework for access control policy specification and enforcement. *J Syst Archit*. 2011;57(4):412-424.
39. Rubio-Medrano CE, D'Souza C, Ahn G-J. Supporting secure collaborations with attribute-based access control. In: Proceedings of the 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom); 2013; Austin, TX.
40. Servos D, Osborn SL. HGABAC: towards a formal model of hierarchical attribute-based access control. In: Proceedings of the International Symposium on Foundations and Practice of Security; 2014; Montreal, Canada.
41. Yuan E, Tong J. Attributed based access control (ABAC) for web services. In: Proceedings of the IEEE International Conference on Web Services (ICWS); 2005; Orlando, FL.
42. Shen H, Hong F. An attribute-based access control model for web services. In: Proceedings of the 7th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT); 2006; Taipei, Taiwan.
43. Shu J, Shi L, Xia B, Liu L. Study on action and attribute-based access control model for web services. In: Proceedings of the 2nd International Symposium on Information Science and Engineering (ISISE); 2009; Shanghai, China.
44. Lang B, Li H, Ni W. Attribute-based access control for layered grid resources. In: Proceedings of the International Conference on Future Generation Communication and Networking; 2010; Jeju Island, South Korea.
45. Shen H. A semantic-aware attribute-based access control model for web services. In: Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing; 2009; Taipei, Taiwan.
46. Kerschbaum F. An access control model for mobile physical objects. In: Proceedings of the 15th ACM Symposium on Access Control Models and Technologies; 2010; Pittsburgh, PA.
47. Buehrer DJ, Wang C-Y. CA-ABAC: class algebra attribute-based access control. In: Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology; 2012; Macau, China.
48. Liang F, Guo H, Yi S, Ma S. A multiple-policy supported attribute-based access control architecture within large-scale device collaboration systems. *J Netw*. 2012;7(3):524.
49. Burmester M, Magkos E, Chrissikopoulos V. T-ABAC: an attribute-based access control model for real-time availability in highly dynamic systems. In: Proceedings of the Symposium on Computers and Communications (ISCC); 2013; Split, Croatia.
50. Smari WW, Clemente P, Lalande J-F. An extended attribute based access control model with trust and privacy: application to a collaborative crisis management system. *Future Gener Comput Syst*. 2014;31:147-168.
51. Zhang YS, Wu MF, Wu L, Li YY. Attribute-based access control security model in service-oriented computing. In: Proceedings of the 2012 International Conference on Cybernetics and Informatics; 2014; Chongqing, China.
52. Sahai A, Waters B. Fuzzy identity-based encryption. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques; 2005; Aarhus, Denmark.
53. Zhu Y, Hu H, Ahn G-J, Huang D, Wang S. Towards temporal access control in cloud computing. In: Proceedings of the 31th Annual IEEE International Conference on Computer Communications (INFOCOM); 2012; Orlando, FL.
54. Yan Z, Li X, Wang M, Vasilakos AV. Flexible data access control based on trust and reputation in cloud computing. *IEEE Trans Cloud Comput*. 2017;5(3):485-498.
55. Lee C-C, Chung P-S, Hwang M-S. A survey on attribute-based encryption schemes of access control in cloud environments. *Int J Netw Secur*. 2013;15(4):231-240.

56. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security; 2006; Alexandria, VA.
57. Yu S, Wang C, Ren K, Lou W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In: Proceedings of the IEEE INFOCOM; 2010; San Diego, CA.
58. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: Proceedings of the IEEE Symposium on Security and Privacy (SP '07); 2007; Berkeley, CA.
59. Yang K, Liu Z, Jia X, Shen XS. Time-domain attribute-based access control for cloud-based video content sharing: a cryptographic approach. *IEEE Trans Multimed*. 2016;18(5):940-950.
60. Fan K, Wang J, Wang X, Li H, Yang Y. A secure and verifiable outsourced access control scheme in fog-cloud computing. *Sensors*. 2017;17(7):1695.
61. Bobba R, Khurana H, Prabhakaran M. Attribute-sets: a practically motivated enhancement to attribute-based encryption. In: Proceedings of the European Symposium on Research in Computer Security; 2009; Saint-Malo, France.
62. Cheung L, Newport C. Provably secure ciphertext policy ABE. In: Proceedings of the 14th ACM Conference on Computer and Communications Security; 2007; Alexandria, VA.
63. Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures. In: Proceedings of the International Conference on Applied Cryptography and Network Security; 2008; New York, NY.
64. Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. In: Proceedings of the International Colloquium on Automata, Languages, and Programming; 2008; Reykjavik, Iceland.
65. Emura K, Miyaji A, Nomura A, Omote K, Soshi M. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In: Proceedings of the International Conference on Information Security Practice and Experience; 2009; Xi'an, China.
66. Ibraimi L, Petkovic M, Nikova S, Hartel P, Jonker W. Mediated ciphertext-policy attribute-based encryption and its application. In: Proceedings of the International Workshop on Information Security Applications; 2009; Busan, South Korea.
67. Ibraimi L, Tang Q, Hartel P, Jonker W. Efficient and provable secure ciphertext-policy attribute-based encryption schemes. In: Proceedings of the International Conference on Information Security Practice and Experience; 2009; Xi'an, hina.
68. Liang X, Cao Z, Lin H, Xing D. Provably secure and efficient bounded ciphertext policy attribute based encryption. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security; 2009; Sydney, Australia.
69. Lewko A, Okamoto T, Sahai A, Takashima K, Waters B. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques; 2010; French Riviera.
70. Li J, Shi Y, Zhang Y. Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage. *Int J Commun Syst*. 2017;30(1):e2942.
71. Jiang Y, Susilo W, Mu Y, Guo F. Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing. *Future Gener Comput Syst*. 2018;78:720-729.
72. Cui H, Deng RH, Lai J, Yi X, Nepal S. An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited. *Computer Networks*. 2018;133:157-165.
73. Ning J, Cao Z, Dong X, Liang K, Ma H, Wei L. Auditable σ time outsourced attribute-based encryption for access control in cloud computing. *IEEE Trans Inf Forensics Secur*. 2018;13(1):94-105.
74. Wang G, Liu Q, Wu J. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In: Proceedings of the 17th ACM Conference on Computer and Communications Security; 2010; Chicago, IL.
75. Horwitz J, Lynn B. Toward hierarchical identity-based encryption. In: International conference on the theory and applications of cryptographic techniques; 2002; Amsterdam, The Netherlands.
76. Wan Z, Liu J, Deng RH. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE Trans Inf Forensics Secur*. 2012;7(2):743-754.
77. Jung T, Li X-Y, Wan Z, Wan M. Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. *IEEE Trans Inf Forensics Secur*. 2015;10(1):190-199.
78. Liu JK, Au MH, Huang X, Lu R, Li J. Fine-grained two-factor access control for web-based cloud computing services. *IEEE Trans Inf Forensics Secur*. 2016;11(3):484-497.
79. Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 14th ACM Conference on Computer and Communications Security; 2007; Alexandria, VA.
80. Birrell E, Schneider FB. Federated identity management systems: a privacy-based characterization. *IEEE Secur Priv*. 2013;11(5):36-48.
81. Chen J, Wu G, Ji Z. Secure interoperation of identity managements among different circles of trust. *Comput Stand Interfaces*. 2011;33(6):533-540.
82. Jiang J, Duan H, Lin T, Qin F, Zhang H. A federated identity management system with centralized trust and unified single sign-on. In: Proceedings of the 6th International ICST Conference on Communications and Networking in China (CHINACOM); 2011; Harbin, China.
83. Bhonsle MV, Poolsappasit N, Madria SK. Etis–efficient trust and identity management system for federated service providers. In: Proceedings of the 27th International Conference on Advanced Information Networking and Applications (AINA); 2013; Barcelona, Spain.
84. Mármol FG, Girao J, Pérez GM. TRIMS, a privacy-aware trust and reputation model for identity management systems. *Computer Networks*. 2010;54(16):2899-2912.

85. Kanwal A, Masood R, Shibli MA. Evaluation and establishment of trust in cloud federation. In: Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication; 2014; Siem Reap, Cambodia.

86. Alguliev RM, Abdullayeva FC. Identity management based security architecture of cloud computing on multi-agent systems. In: Proceedings of the 3rd International Conference on Innovative Computing Technology (INTECH); 2013; London, UK.

87. Chadwick DW, Inman G, Coxwell P. CardSpace in the cloud. In: Proceedings of the 17th ACM Conference on Computer and Communications Security; 2010; Chicago, IL.

88. Khattak ZA, Sulaiman S, Ab Manan J-L. A study on threat model for federated identities in federated identity management system. *Int Symp Inf Technol*. 2010;2:618-623.

89. Chadwick DW, Inman G. The trusted attribute aggregation service (TAAS)-providing an attribute aggregation layer for federated identity management. In: Proceedings of the International Conference on Availability, Reliability and Security; 2013; Regensburg, Germany.

90. Samlinson E, Usha M. User-centric trust based identity as a service for federated cloud environment. In: Proceedings of the 4th International Conference on Computing, Communications and Networking technologies (ICCCNT); 2013; Tiruchengode, India.

91. Gao H, Yan J, Mu Y. Dynamic trust model for federated identity management. In: Proceedings of the 4th International Conference on Network and System Security (NSS); 2010; Melbourne, Australia.

92. Coyne E, Weil TR. ABAC and RBAC: scalable, flexible, and auditable access management. *IT Professional*. 2013;15(3):14-16.

93. Fall D, Blanc G, Okuda T, Kadobayashi Y, Yamaguchi S. Toward quantified risk-adaptive access control for multi-tenant cloud computing. In: Proceeding of the 6th Joint Workshop on Information Security; 2011; Kaohsiung, Taiwan.

94. Sakai H. Standardization activities for cloud computing. *NTT Tech Rev*. 2011;9(6):1-6.

95. Willcocks L, Venters W, Whitley E. *Moving to the Cloud Corporation: How to Face the Challenges and Harness the Potential of Cloud Computing*. London, UK: Springer; 2013.

96. Cai F, Zhu N, He J, Mu P, Li W, Yu Y. Survey of access control models and technologies for cloud computing. *Cluster Computing*. 2018:1-12.

97. Khan AR, Othman M, Madani SA, Khan SU. A survey of mobile cloud computing application models. *IEEE Commun Surv Tutor*. 2014;16(1):393-413.

98. Alizadeh M, Abolfazli S, Zamani M, Baharun S, Sakurai K. Authentication in mobile cloud computing: a survey. *J Netw Comput Appl*. 2016;61:59-80.

How to cite this article: El Sibai R, Gemayel N, Bou Abdo J, Demerjian J. A survey on access control mechanisms for cloud computing. *Trans Emerging Tel Tech*. 2019;e3720. https://doi.org/10.1002/ett.3720