Al-Rafidain Engineering Journal (AREJ) Vol. 30, No. 1, March 2025, pp. 71-90

Literature Review on Access Control Models in Software Architecture

Ali Sohofi
sohofi@shdu.ac.ir

Amir Dehbashi Dezfouli
amir.dehbashi@shdu.ac.ir

Computer Engineering Department, Shahab Danesh University, Qom, Iran

Received: May 1st, 2024 Received in revised form: July 1st, 2024 Accepted: July 28th, 2024

ABSTRACT

In the realm of software architecture, ensuring the security of organizational assets against unauthorized access is crucial for cybersecurity. This paper conducts a comprehensive exploration of access control, traversing from traditional paradigms like DAC, MAC, RBAC, and ABAC to contemporary trends such as policy-based access control and blockchain-based access control. We categorize access control models based on their usage, considering new emerging fields like cloud computing and the Internet of Things (IoT). Finally, we acknowledge the ongoing challenges in access control and propose promising directions for future research, including the integration of blockchain and the potential of machine learning to enhance access control mechanisms.

Keywords:

Access Control; Access Control Models; Access Policy; Software Architecture; Security.

https://rengj.mosuljournals.com
Email: alrafidain_engjournal1@uomosul.edu.iq

==========

1. INTRODUCTION

Modern software architectures necessitate robust security mechanisms to protect sensitive data and critical organizational assets from unauthorized access. Achieving optimal information security requires a well-defined balance between stringent access control policies and efficient user authorization workflows. This equilibrium is facilitated by access control models, which provide a systematic approach to managing resource accessibility based on user privileges and security principles.

Access control, a linchpin in information security technologies, meticulously governs requests for security access to vital system resources [1]. The fundamental purpose of access control is to establish a robust security layer, shielding the crucial data of diverse organizations [2]. This mechanism meticulously grants access solely to authorized users within organizations, steadfastly denying entry to unauthorized users and external entities [3].

Embarking on a comprehensive exploration, this paper delves into the intricate domain of access control, unveiling its foundational concepts, structural hierarchies, and the diverse array of evolving models. Our journey initiates with an in-depth examination of foundational access control models, traversing traditional paradigms from discretionary access control (DAC) [4], where resource owners exercise discretion, to mandatory access control (MAC) [5], imposing access policies through system-wide security measures. Building upon this groundwork, we progress into the revolutionary paradigm of role-based access control (RBAC) [6], streamlining user management in complex organizational structures.

Amidst our survey, our focus extends to cutting-edge advancements reshaping access control paradigms. A cornerstone in modern cybersecurity, Attribute-Based Access Control (ABAC) [7] facilitates nuanced and fine-grained control by considering multiple users and resource attributes. We navigate the complexities of ABAC, uncovering its potential to mitigate limitations inherent in traditional models, particularly within dynamic and evolving environments. Furthermore, our exploration

extends to emerging trends, including policy-based access control and blockchain-based access control, unraveling innovative solutions and their implications for the future of cybersecurity practices in complex environments like cloud computing and the Internet of Things.

This survey presents traditional and modern access control models categorized by their usage. For each model, some details about its framework are presented. We also discuss the current challenges of access control models and the road ahead. The remainder of the paper is structured as follows: We present the principles of access control in Section II. Access control models are reviewed and categorized in Section III, followed by a discussion about features critical for the access control model in Section IV. We identify open issues and draw research directions for future research in Section V.

2. ACCESS CONTROL PRINCIPLES

Access control involves three stages: identification, authentication, and authorization [8]. The requesting users are identified and evaluated by the existing system. After their individual authentication, based on the level of privileges defined and specified by the organization, they are allowed to access the desired resource according to these decisions.

A. Identification

Initiating the access control process, identification involves users asserting their individual identities, validated through diverse credentials such as user ID, process ID, or smart cards. The uniqueness of these credentials is paramount for effective differentiation between end users within a given system.

B. Authentication

When the initial identification of a user is completed through an identity credential, the authentication process begins. In this phase, the user must demonstrate that the claim of being the person associated with the provided identity is accurate. To substantiate this claim, the user undergoes a credential verification process facilitated by the organizational mechanism. If the user successfully completes this process, their identity is verified. With a certain level of confidence, the authentication mechanism establishes that the user possesses the provided credentials and has complete control over them. Various credentials, such as passwords, PIN codes, smart cards, and biometric data, can be utilized to verify the individual identity of users.

Authentication factors are user credentials that prove identity for accessing protected resources. These factors fall into three categories [3]:

- Knowledge-based: Involves private information like passwords or PINs.
- Possession-based: Utilizes physical objects such as smart cards.
- Biometric-based: Validates physical or behavioral traits like fingerprints.

The levels of authentication implemented and enforced by organizations hinge on the confidentiality of their sensitive information and data. Some organizations aim to establish a robust and highly secure system, while others opt for a more straightforward approach. As described in the previous section, the system's authentication factors dictate the level of authentication the organization's mechanism adopts. A direct and inseparable relationship exists between authentication factors and levels. Generally, authentication levels are categorized into three types: single-factor, two-factor, and multi-factor. These categories serve as benchmarks, where a higher number corresponds to a higher level of security for the system.

C. Authorization

Authorization is the third and final stage in the access control process, following identification and authentication. Once a user’s identity has been established and verified through the initial stages, authorization comes into play to determine what the authenticated user is allowed to do within the system. This critical step ensures that users can only access the resources and perform the actions they have been explicitly permitted to, thereby protecting sensitive data, maintaining system integrity, and ensuring compliance with organizational policies and regulatory requirements. Without proper authorization, even authenticated users could inadvertently or maliciously access, modify, or delete data they are not entitled to, leading to potential security breaches, data loss, and other adverse impacts on the organization. Therefore, implementing a robust authorization mechanism is essential for safeguarding the system against unauthorized access and ensuring that every user interaction aligns with the organization's security and operational policies.

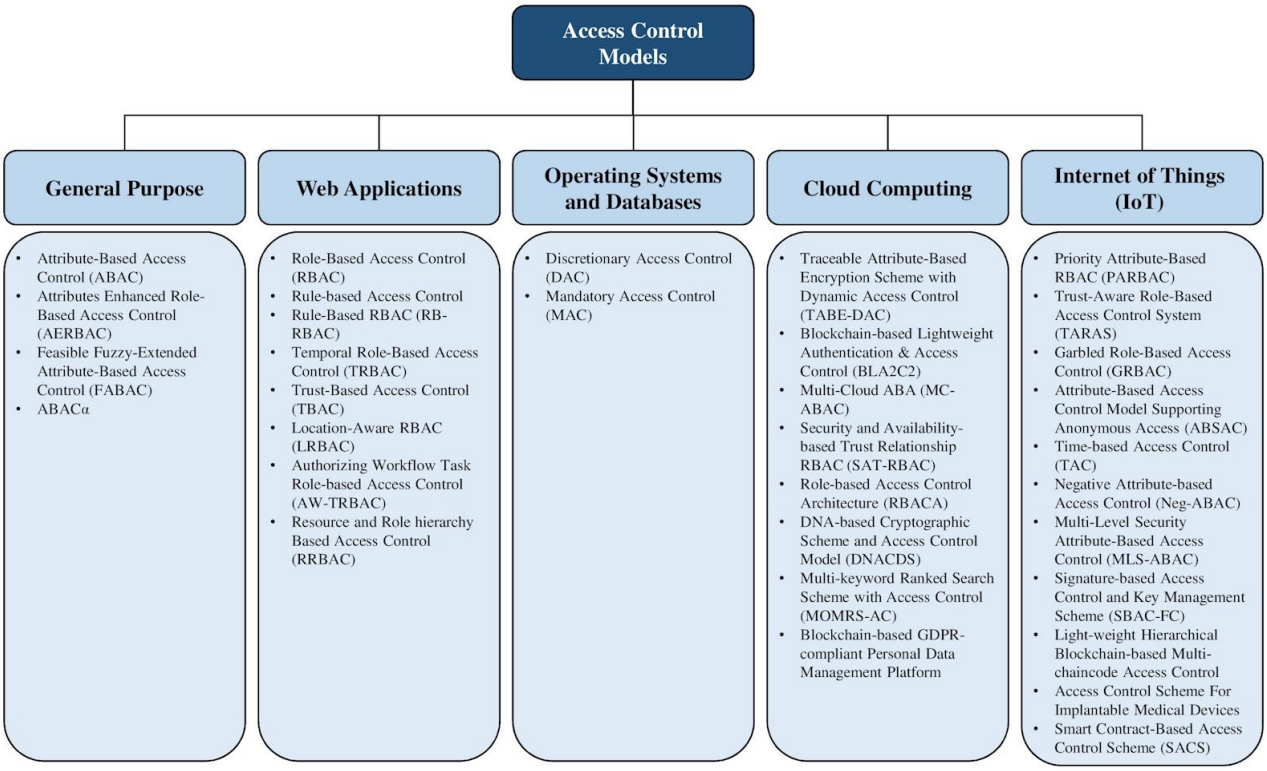

Fig. 1 Categories of access control models based on their application

3. ACCESS CONTROL MODELS

Access control models articulate the policies governing resource access management within organizations, manifesting in a plethora of designs, constituent elements, rules, and privacy considerations. These models, each crafted for specific scenarios, present distinct advantages and disadvantages [9]. Organizations, dictated by their unique data and circumstances, carefully select an access control model aligning with their specific needs. When existing models fall short, a hybrid approach employing multiple models becomes imperative to forge a more robust system. Consequently, organizations face no constraints in utilizing only existing models [10].

Numerous access control models have been proposed, often classified in diverse ways in [11], [12], [13], [14]. These models exhibit versatility, finding applications in various domains, with some tailored to specific use areas. This paper adopts a categorization based on usage, encompassing general-purpose, web applications, operating systems, databases, cloud computing, and the Internet of Things, see Figure 1. It is essential to note that demarcating a clear boundary between these categories is challenging, and our classification is based on existing literature. A model's placement in a category does not preclude its potential application in other contexts. This section delves into each category, providing a detailed exploration of the distinct applications within these classifications.

3.1. General Purpose

Attribute-Based Access Control (ABAC): This model offers fine-grained access control, flexibility, and dynamicity by leveraging attributes allocated by attribute authorities [7]. A Boolean formula defines access control policies using attribute sets, eliminating the need for creating numerous roles or access control lists. ABAC employs attributes like citizenship, IP address, identity, location, and username, evaluating them against objects, subjects, environment, and operations rules. Dynamic behavior allows real-time detection of changes in attribute values or user identities, a capability lacking in the RBAC model. However, ABAC faces complexity challenges as the number of attributes increases [15]. User attributes, defined by administrators, include name, designation, organizational affiliation, gender, age, nationality, or security clearance, requiring regular management during personnel changes [16].

ABAC's functionality relies on device policies, procedural rules, or documents. For instance, a physician's exclusive access to patient

records in a medical emergency illustrates the model's specific access privileges for subjects [17]. ABAC safeguards objects based on defined policies and attributes, requiring the access control mechanism (ACM) to intelligently process information, policies, attributes, and their chronology for decision-making [18]. Least privilege ensures users access only necessary resources, dynamic behavior automates operations, safety prevents permission leakage, separation of duties limits access, capability delegation allows users to revoke granted features, configuration flexibility streamlines installation, and auditing monitors user requests. ABAC provides the most granular control over access decisions. It leverages a combination of factors, including user attributes (e.g., job title, location), resource attributes (e.g., data classification, creation time), and environmental factors (e.g., time of day), to determine access permissions. This approach enables highly dynamic and secure access control. However, the power of ABAC comes with a price. Defining and managing a vast number of attributes necessitates meticulous planning and ongoing maintenance. The complexity of implementation can pose a significant challenge for some organizations. Figure 2 depicted an example scenario of ABAC [17].

Attributes Enhanced Role-Based Access Control (AERBAC): Rajpoot et al. proposed a hybrid access control model that integrates key aspects of both role-based and attribute-based access control. In contrast to the ABAC model, this hybrid model establishes a comprehensive framework for evaluating a subset of rules based on user roles. Simultaneously, it preserves the advantages inherent in the RBAC model, such as role assignment to users, scrutiny of permissions linked to user roles, and simplifying system management complexities in diverse large-scale organizations [19], [20].

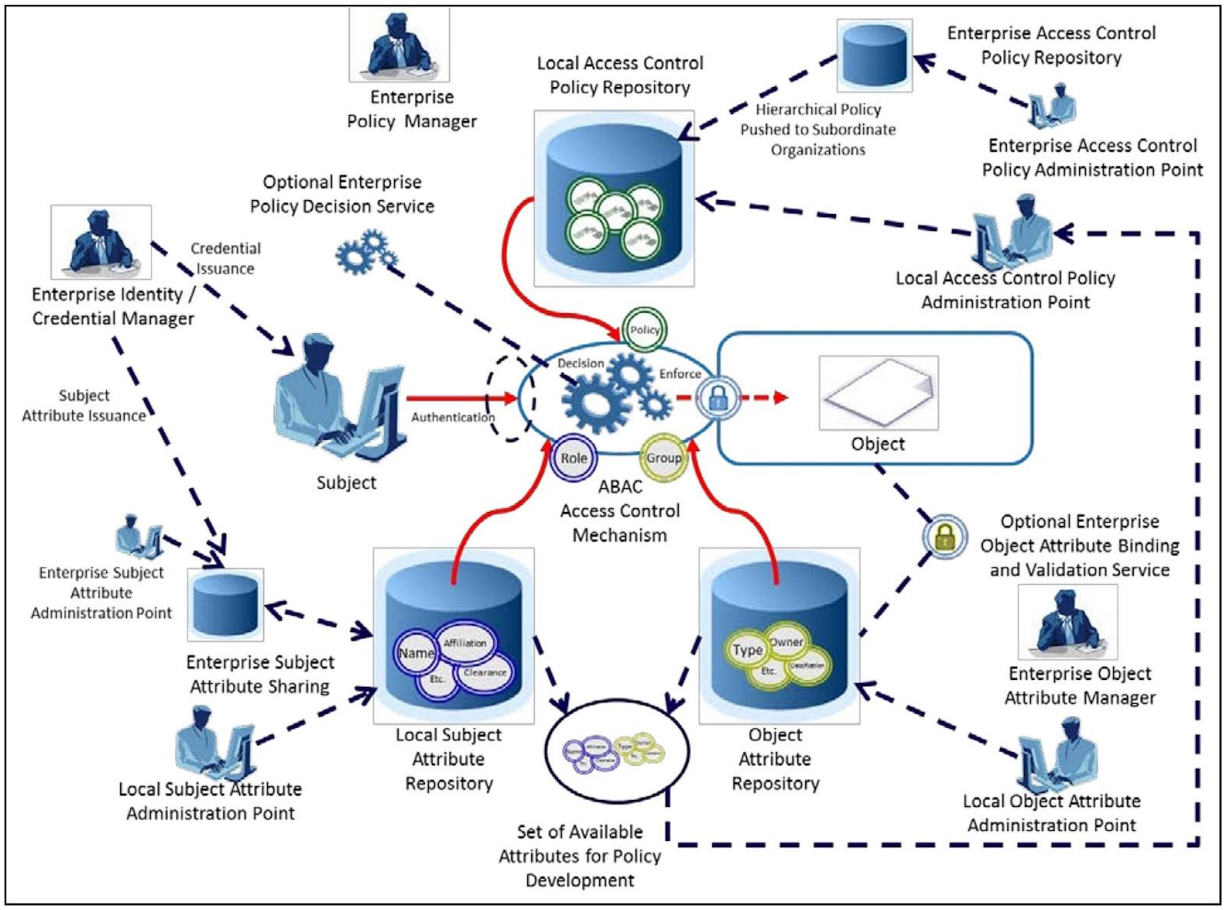

Fig. 2 An example scenario of ABAC access control [17]

Feasible Fuzzy-Extended Attribute-Based Access Control (FABAC): is an extended model of attribute-based access control, aims to efficiently and flexibly handle critical authorizations. It surpasses the ABAC model by optimizing resource utilization and aligning with business requirements. The model has undergone testing in high-risk applications, specifically in

audit mechanisms and credit systems, evaluating aspects such as risks, usability, analysis, and effectiveness. The FBAC model has demonstrated superior time efficiency and flexibility in various tests compared to ABAC [21]. However, it falls short of providing stringent security measures and adhering to the principle of least privilege [22].

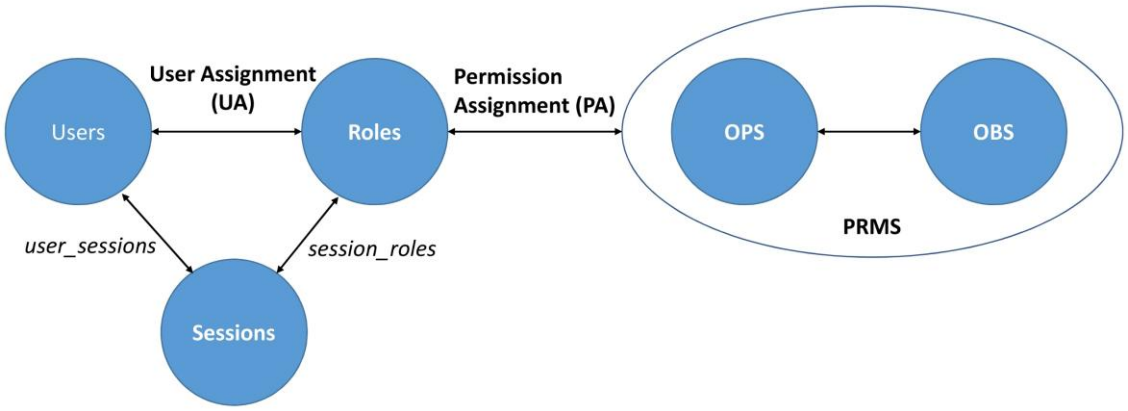

Fig. 3 Core RBAC adopted from [24]

ABACα: The model aims to create an ABAC model with the "just sufficient" features, allowing it to be easily and naturally configured for DAC, MAC, and RBAC functionalities [23]. The key elements of the unified ABACα model include users (U), subjects (S), objects (O), user attributes (UA), subject attributes (SA), object attributes (OA), permissions (P), authorization policies, and policies for constraint checking to create and modify subject and object attributes. Attributes serve as functions mapping an entity to a specific value within its range. While security administrators create user attributes, users generate attributes for subjects during their creation, and the same applies to object attributes. Permissions represent the privileges a user can possess on objects, exercised through a subject. An authorization policy for a specific permission involves a subject, an object, and returns either true or false based on attribute values [23].

3.2. Web Applications

Role-Based Access Control (RBAC): This model revolutionizes access control by abstracting permissions into roles and associating users with these roles. This hierarchical structure simplifies administrative tasks, particularly in dynamic settings where user roles frequently change. RBAC excels in web server applications, consolidating authorization data into a unified RBAC authorization database. RBAC adheres to the principles of least privilege and separation of duty. The least privilege ensures users receive only the necessary privileges for their roles, mitigating inadvertent security risks. Separation of duty partitions tasks and privileges among roles, enhancing overall system security. This model, centered around roles, comprises five entities: objects (resources), operations (authorized activities), permissions (object-operation combinations), roles (sets of permissions), and subjects (users). Users access resources exclusively through their assigned roles, streamlining security management and reducing operational complexity [6].

RBAC's hierarchical design introduces role hierarchy, reflecting organizational structures and simplifying role and permission management. The model incorporates constraints to enforce higher-level organizational policies, adding flexibility. RBAC's modular structure includes three key modules: basic RBAC, hierarchical RBAC, and limited RBAC. These modules contribute to finer control, adaptability, and addressing challenges such as scalability and dynamic role changes [24]. RBAC is a versatile access control model, offering a nuanced approach to security in complex organizational settings. Its abstract design, adherence to security principles, and modular structure make it a valuable asset, balancing organizational control with user flexibility [25]. RBAC offers a well-balanced approach between security and manageability. It simplifies access control by defining permissions for predefined roles within the system. Users are then assigned roles that align with their job functions or responsibilities. This method fosters greater scalability and manageability compared to DAC, particularly in large organizations with extensive user bases. However, RBAC may lack the granularity of DAC. If specific access requirements arise that fall outside the purview of existing roles, additional roles may need to be created, potentially increasing management overhead.

Core RBAC model element sets and relations are depicted in Figure 3 [24].

Rule-based Access Control: The rule-based access control model is applied in the context of Web-based social networks, facilitating access to online resources. Within this framework, authorized subjects are defined according to the relational form, depth, and degree of trust among network users using attribute-based RBAC. Resource access is granted based on specific access rules. In rule-based models, resource owners specify protocols and outline the profile of authorized users through one or more access conditions. These conditions encompass restrictions on the type, depth, and trust level of their connections with other network users. The access control requires a specific object, articulated through a series of conditions [26].

Rule-Based RBAC (RB-RBAC): RB-RBAC is an extended model of role-based access control designed to introduce dynamic behavior into the system. This model introduces a new set of rules for defining access policies, which are automatically triggered for user-role assignments. Changes are implemented in the section responsible for assigning roles to users. The system, as designed, initially checks the characteristics of users and roles. If the characteristics on both sides match their corresponding values, automatic allocation occurs; otherwise, it does not [27].

Temporal Role-Based Access Control (TRBAC): TRBAC extends role-based access control, enabling dynamic behavior and removing temporal restrictions on role activation. It manages temporary roles, introducing transient dependencies through role triggers [28]. These triggers limit the activation of specific roles within defined time frames, allowing users to set periods for immediate or delayed role switching. TRBAC overcomes non-permanent limitations, addressing seasonal role changes and dependencies. Users can invoke triggers to instantly or delayed switch roles, facilitating dispute resolution. Additionally, security officers can enhance emergency response capabilities by manipulating role states, and user controls through runtime requests, providing immediate or delayed actions for scenarios like temporarily preventing a user from activating a potentially harmful role [29].

Trust-Based Access Control (TBAC): This model addresses elevated threat levels in online social networks (OSN) where users engage in data interactions, posing security risks. The model introduces roles like owner, contributor, and stakeholder, each associated with specific security levels [30]. A multi-role environment enables users to apply multiple security parameters, allowing them and their friends to make access decisions without policy conflicts. However, the TBAC model proposed for OSNs lacks suitability for other domains like wireless sensor networks, IoT, and cloud computing [31]. Additionally, the absence of an administrator role raises concerns about addressing security issues, such as removing unethical content. The model lacks real social network simulation to validate its effectiveness against other methods [32].

Location-Aware RBAC (LRBAC): LRBAC extends the role-based access control model to enhance user access reliability and security by incorporating real-time location checks. It prevents users from accessing undesired geographic locations by leveraging location information. This model is particularly suitable for applications involving both static and dynamic objects, where considering the geographic location of users and objects is crucial for access decisions [33]. LRBAC has been expanded to incorporate Location-Based Services (LBS) concepts. Various levels of granularity can be used to specify locations. Access to resources is determined by the user's role, which also highlights constraints imposed by role-based entities on access control, encompassing dynamic separation of duties and role hierarchy.

Authorizing Workflow Task Role-based Access Control (AW-TRBAC): AW-TRBAC is a dynamic access control model rooted in role-based access control, designed to address limitations in access management within a borderless network environment. The model innovatively combines RBAC and TBAC to establish an access control approach based on authorized workflow task roles. It introduces real-time segregation of dynamic tasks in decision-making and policy implementation [34]. The model integrates existing task and workflow concepts, ensuring access governance and responsiveness by considering workload constraints. AW-TRBAC improves upon traditional access control models, like role-based access control, by dynamically granting access rights to users and incorporating dynamic segregation of duties (SoD) and process workflow.

Resource and Role hierarchy Based Access Control (RRBAC): This is an extension of role-based access control that simplifies permission assignment by introducing a resource hierarchy-based authorization model. Unlike policy-based solutions, it utilizes a resource decision tree for efficient license-related operations. Experiments demonstrate RRBAC's

effectiveness in license allocation, validation, and cancellation. Permissions are directly assigned to data, with inheritance for hierarchically organized data. RRBAC integrates data permissions management with user/role ACL permissions, particularly for smaller teams or personal projects. Resource owners retain granular control over access permissions, fostering ease of use and customization. However, this user-centric approach can introduce security vulnerabilities if

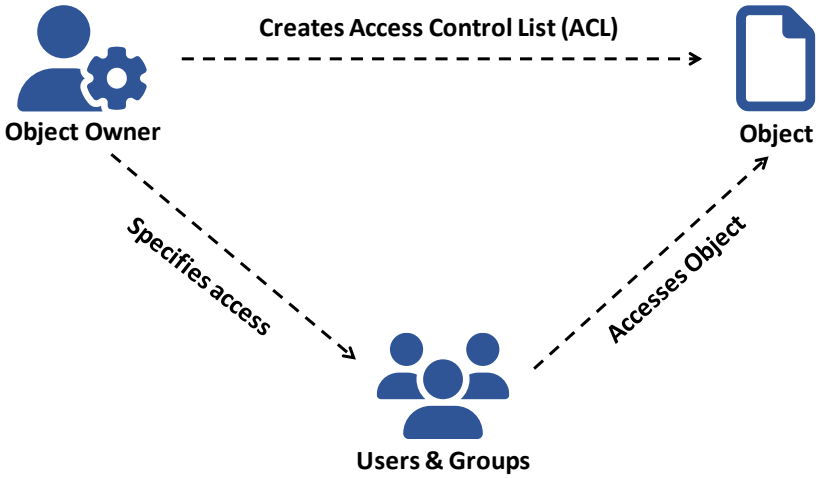

Fig. 4 Discretionary Access Control (DAC)

enhancing file system permissions through centralized control [35]. While aiming to improve policies and efficiency, RRBAC introduces complexity to permission validation, particularly for large organizations, despite streamlining permission assignment.

3.3. Operating Systems and Databases

Discretionary Access Control (DAC): The DAC is a model that operates on owner-based access principles, where the creator of a resource or object serves as the owner. In this model, the owner holds the authority to determine access policies for subjects or users. Notably, there is no reliance on administrators to manage access rights in this context. DAC is further divided into two types: liberal DAC and strict DAC. Under liberal DAC, the owner has the flexibility to transfer access rights or ownership to others, allowing them to function as resource owners. Conversely, in strict DAC, access rights are confined to the original owner, with no provision for ownership transfer [4], [6]. The DAC model operates based on the owner's discretion, with access control policies enforced across three categories: resource ownership, user identities, and permission delegation. However, DAC exhibits limitations that make it unsuitable for commercial and government organizations. These limitations stem from the model's allowance for users to define access rights, posing risks such as Trojan horse attacks [36]. DAC excels in its straightforward implementation, access control lists (ACLs) are not meticulously managed. In large-scale deployments with numerous users and resources, maintaining comprehensive ACLs can become cumbersome, potentially leading to inadvertent misconfigurations or unauthorized access. Figure 4 shows the core concept of DAC.

Mandatory Access Control (MAC): MAC determines access to resources using a hierarchical structure. It manages user or process access rights to system resources by assigning security levels to users and security labels to objects. Users are associated with security levels, and their access is limited to resources with security levels equal to or lower than their own [37]. The administrator strictly controls access rights and sets permissions, making MAC effective for military and commercial systems due to its high-level security [7]. In the MAC model, a centralized mechanism is employed to permit or deny access to resources, contributing to its enhanced security, flexibility, and efficiency for commercial and military use. Also known as lattice-based access control [5] or multilevel security, MAC involves classifications (e.g., top secret, secret, confidential, classified) and categories, forming a partially ordered lattice of security levels. This model uses a hierarchical network of security tags, allowing information security specialists to control users' access rights by assigning security labels to resources and security levels to users. Access is granted based on the match or hierarchy of security levels,

following the Bell-Lapadula mathematical model [38]. This model functions as a computer security policy state transfer machine, defining access control rules and security label assignments from the most sensitive to the least sensitive level.

confidentiality by tracing malicious users through accountability measures, preventing unauthorized sharing of secret data on the cloud. However, the efficiency of the master-slave blockchain architecture used in TABE-DAC is limited due to

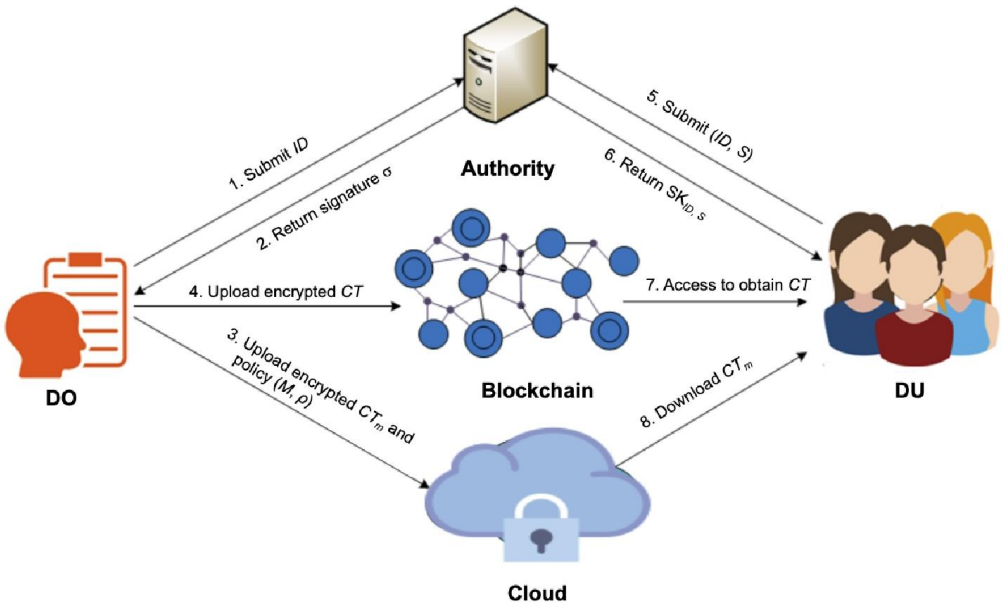

Fig. 5 System model of Traceable Attribute-Based Encryption Scheme with Dynamic Access Control (TABE-DAC) [39]

However, MAC has limitations, including difficulties in system management, especially as the system size increases, and the high cost and operational challenges associated with the dependence on trusted components. MAC prioritizes robust security through predefined security labels that dictate access privileges for both users and resources. This strict approach is ideally suited for safeguarding highly sensitive information in government or military environments. Conversely, the inflexibility of MAC can hinder collaboration. The complexity of establishing and managing security labels can be significant, and users might encounter limitations in accessing resources crucial for collaboration, potentially impacting workflow efficiency.

3.4. Cloud Computing

Traceable Attribute-Based Encryption Scheme with Dynamic Access Control (TABE-DAC): TABE-DAC is an efficient model that leverages attribute-based encryption and blockchain technology to share secret data on cloud servers. Unlike traditional attribute-based encryption, TABE-DAC provides dynamic access control, allowing data owners to modify access policies flexibly [39]. This model enhances the necessity for cross-chain collaboration in data storage and query. Figure 5 illustrates the system model of TABE-DAC [39].

Blockchain-based Lightweight Authentication & Access Control (BLA2C2): BLA2C2 [40] is a novel blockchain-based lightweight authentication and access control layer for dynamic cloud deployments. The model addresses common attacks on cloud deployments, such as brute force, masquerading, improper access, and session hijacking. Existing models are criticized for being either complex or lacking security under multiple attacks, limiting their real-time applicability. The proposed model includes a header-level lightweight sanitization layer to remove various data-level attacks, a lightweight authentication layer using IP matching and reverse geolocation mapping, and an efficient blockchain-based access control model integrated with Grey Wolf Optimization for faster response. The design aims to overcome complexities and inflexibility observed in current authentication and access control models for dynamic cloud scenarios.

Multi-Cloud ABAC (MC-ABAC): MCABAC [41] is an extension of the Attribute Based Access Control (ABAC) model, tailored

for secure collaboration and cross-tenant access in a multi-cloud environment. This model incorporates key entities such as tenants, cloud customers, and cloud service providers, introducing multiple trust relations to facilitate collaboration and resource sharing across different clouds. The model leverages the flexibility and adaptability of ABAC to accommodate diverse access control models used by various cloud providers, making it suitable for controlling access to different cloud services, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), or Software as a Service (SaaS).

Security and Availability-based Trust Relationship RBAC (SAT-RBAC): Authors in [42] proposed a novel approach to Role-Based Access Control (RBAC) in cloud environments. The model assesses the trust relationship between users and roles based on three elements: the security situation of the host, network availability, and server protection. The computed trust degree falls into three zones: unbelievable, probable, and believable. Authorization decisions are made accordingly, with the model incorporating a Bayesian Probability Distribution in the probable believable zone. The SAT-RBAC model is designed to address the dynamic and ad hoc nature of relationships between resources and users in cloud environments, offering a security-based scheduling model to enhance trust assessment.

Role-based Access Control Architecture (RBACA): RBACA [43] is an RBAC architecture for a multi-domain cloud environment. In this architecture, service providers and domain administrators are responsible for access control and policy management, respectively. While there is some isolation between service providers and domains, service providers are allowed to modify domain policies, leading to an increased management burden for domain administrators. When users request cross-domain access, domain administrators send policies to corresponding domains without considering policy differences. The paper proposes a scalable RBAC architecture for cloud computing, emphasizing a flexible framework for policy management in a multi-domain environment. The architecture is based on traditional access control and the RBAC model, using the XACML policy language to demonstrate access control policies. The goal is to provide a standard approach for managing resources and services in a general multi-domain cloud environment.

DNA-based Cryptographic Scheme and Access Control Model (DNACDS): A novel DNA-based Cryptographic Scheme and Access Control Model (DNACDS) is proposed in [44] to address security and access control challenges in cloud-based Internet of Everything (IoE) applications managing big data. The inadequacy of traditional cryptographic systems based on binary data for IoE big data security is addressed by leveraging DNA cryptography, a bio-inspired approach. DNACDS incorporates a security framework utilizing DNA computing principles and a Station-to-Station Key Agreement Protocol (StS KAP) to generate a robust 256-bit secret key for secure data encryption. An access control mechanism further enhances security by ensuring that only authorized users can access the encrypted data. The DNACDS's contributions include improved security through DNA cryptography, robust access control, and efficiency compared to existing techniques, making it a promising approach for securing big data and access control in cloud-based IoE systems.

Multi-keyword Ranked Search Scheme with Access Control (MOMRS-AC): Guo et al. [45] introduced MOMRS-AC, a secure multi-keyword ranked search scheme with access control for the multi-owner model in cloud computing. MOMRS-AC addresses the limitations of existing schemes by utilizing a symmetric secure KNN algorithm for efficient search on unstructured data and eliminating the need for a Trusted Third-Party (TTP). It allows data owners to independently generate secure indexes and data users to generate a single trapdoor for searching across all data owners, maintaining constant search time regardless of the number of query keywords. Additionally, MOMRS-AC ensures privacy-preserving access control by implementing access policies that do not contain sensitive information, supporting decentralized authorities and large-universe scenarios. The scheme also enables data users to validate search results by verifying the signatures of data owners and allows lightweight devices to outsource verification or decryption tasks to the cloud server, reducing client-side overhead. Extensive experiments using real-world datasets confirm the efficiency and effectiveness of MOMRS-AC.

Blockchain-based GDPR-compliant Personal Data Management Platform: Authors in [46] proposed a blockchain-based access control system to address the challenges of personal data management outlined in the General Data Protection Regulation (GDPR). The increasing use of digital technologies and the vast amount of personal data they collect necessitate a transparent and GDPR-compliant data management system. The proposed solution leverages blockchain

technology to provide public access to immutable records of user consent regarding data collection and processing by service providers. This transparency allows users to monitor how their data is handled and ensures service providers comply with GDPR.

3.5. Internet of Things

Priority Attribute-Based RBAC (PARBAC): PARBAC [47], an extended role-based access control model, addresses the challenges of large medical scenarios within the Azure Internet of Things cloud network. It introduces a priority-based authentication mechanism to enhance adaptability and flexibility. The model enforces resource access rights based on priority, streamlining operations in large organizations. PARBAC model operates in seven steps, involving token issuance, API calls, Azure resource manager decisions, role-based advice, activity verification, logging restrictions, and access blocking. The prioritization mechanism reduces the operational burden on cloud servers and handles dynamic scenarios in large organizations.

Trust-Aware Role-Based Access Control System (TARAS): Guoak et al. [31] introduced the TARAS as an extended model to enhance the security of IoT device communication. TARAS establishes trust relationships between users, IoT devices, and smart devices by considering users with similar roles to respond similarly. The model effectively identifies unauthorized and malicious users, employs dynamic trust estimation, enhances data integrity, and improves system performance, availability, detection accuracy, and robustness, especially in high attack density scenarios. TARAS utilizes a multidisciplinary approach, incorporating the concept of I-sharing from psychology to quickly establish trust between smart objects and new users without prior interactions. Adaptive, dynamic trust estimation allows TARAS to revoke access rights from malicious users, maximizing system integrity and service availability. Experimental results demonstrate TARAS's effectiveness in detecting malicious or benign users while optimizing system performance through fine-tuned trust threshold settings.

Garbled Role-Based Access Control (GRBAC): This model is a fine-grained security solution addressing security and privacy challenges in data outsourcing for IoT environments. The GRBAC model [48] employs a garbled function and is tailored for organizations where roles are not exposed to servers and users. Its main feature ensures that a user cannot activate more than one garbled role set, and the organization's data and roles remain hidden for privacy maintenance. While the algorithm used in the system is not concealed, the model is specifically designed to prevent the disclosure of roles and data. It offers fine-grained security, utilizing RSA Oblivious-Transfer for role assignment, and integrates Role-Based Access Control and Dynamic Separation of Duty. Despite its advantages, the model has limitations, such as inflexibility, restrictions on server access to user roles, and the inability of the server to maintain records or control the access control system effectively.

Attribute-Based Access Control Model Supporting Anonymous Access (ABSAC): Zhang [49] proposed this model to safeguard user data in IoT applications within smart cities. Traditional Attribute-Based Access Control (ABAC) models are deemed insufficient for large organizations, and the ABSAC model addresses this by supporting anonymous access and enhancing security for user data transactions in public spaces with minimal risk. The ABSAC model utilizes homomorphic attribute-based signatures (HABSs) [50] to strengthen identity-less ABAC, ensuring authorization is not dependent on identity. By not sending subject attributes to the authorization organization, ABSAC reduces the risk of subject identity re-identification, providing a secure, anonymous access framework. ABSAC inherits features like fine-grained access control, flexible policy, and unlimited object types from ABAC, overcoming the traditional ABAC framework's limitation on anonymous access and introducing an audit function to enhance security. Performance tests show that ABSAC's implementation is comparable to ABAC's performance.

Time-based Access Control (TAC): This model secures user data in the Internet of Things (IoT). In TAC [51], user data is categorized into two directional subspaces representing attributes and time generation. Access control and privacy are ensured by encrypting data before transmission, and the data owner/source can grant access using sub-keys. The TAC model efficiently generates sub-keys for each subspace with minimal time and memory space requirements. It proves to be an effective and flexible solution for securely sharing secret data in the IoT environment. The TAC scheme is based on a revised one-way hash chain technique, securing multi-attribute data in IoT by partitioning it into 2-D subspaces based on generation time and data attribute. Data in each subspace is encrypted with corresponding sub-keys for privacy and access control. The TAC model is resource-efficient for IoT applications, particularly those involving real-

time data collection in people-centric scenarios. Experimental results demonstrate its effectiveness, and the model includes improvements to reduce sub-key computation time in various application scenarios.

Negative Attribute-Based Access Control (Neg-ABAC): Aftab et al. [52] presented this model as an extension of the ABAC to restrict unauthorized access by incorporating attributes and negative parameters. This model introduces the concept of negative attributes, enabling the implementation of negative permissions in ABAC. The unique application of Neg-ABAC is demonstrated in the context of the Internet of Medical Things (IoMT) for medical devices. By utilizing negative authorization, the model proves effective in certain scenarios and notably reduces the operational burden on technical teams. The model is distinctive for its different implementation of access control in the IoMT domain. The proposed hybrid model integrates reverse authorization within ABAC, emphasizing the significance of negative attributes in restricting unauthorized users. This model is particularly relevant in scenarios where negative authorization can enhance access control and security measures.

Multilevel Security Attribute-Based Access Control (MLS-ABAC): This model is a scheme designed for secure data access control in the context of the Internet of Things (IoT). It addresses the challenge of providing access control to sensitive data offloaded to the cloud, where devices have varying computational power and security levels. MLS-ABAC [53] efficiently ensures data integrity and multilevel security access control. It proposes a lightweight decryption outsourceable construction using Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to meet the National Institute of Standards and Technology's (NIST) ABAC requirements [17]. The entities involved in the scheme include the Attribute Authority Server (AAS), Identity and Access Management Server (IAMS), Cloud Server (CS), Data Owner (DO), and Data User (DU). AAS generates system parameters and secret keys, IAMS creates tokens based on security levels, CS stores ciphertexts and validates tokens, DO produces sensitive messages with associated security levels, and DU, an IoT device, requests tokens and decrypts ciphertexts. The proposed MLS-ABAC achieves constant ciphertext size and demonstrates efficient encryption and decryption performance. The scheme is shown to be applicable in realistic application scenarios, providing fine-grained access control over encrypted data in IoT environments.

Signature-based Access Control and Key Management Scheme (SBAC-FC): Fog computing, a distributed computing architecture, brings data processing, application functionality, and storage closer to the network's edge, reducing dependency on centralized cloud servers. This proximity to data sources or end-user devices is particularly beneficial for Internet of Things (IoT)-enabled systems in various domains. However, the fog computing-based IoT-enabled system is vulnerable to multiple attacks, necessitating the deployment of security mechanisms like authentication, access control, key management, and malware detection to secure communication. SBAC-FC [54] is a signature-based access control and key management scheme tailored for fog computing-based IoT-enabled big data applications.

Lightweight Hierarchical Blockchain-based Multi-chaincode Access Control: Abdi et al. [55] address the challenges of managing the vast and distributed network of Internet of Things (IoT) devices, which traditional access-control systems struggle to handle due to central authority management issues. The proposed solution introduces a lightweight hierarchical blockchain-based multi-chaincode access control model for improved security and privacy in IoT systems. To overcome scalability issues, the model utilizes a clustering concept with three main components: Edge Blockchain Managers (EBCMs) for local device authentication and authorization, Aggregated Edge Blockchain Managers (AEBCMs) to control different clusters and manage access control policies, and a Cloud Consortium Blockchain Manager (CCBCM) ensuring only authorized users access resources. Smart contracts are employed for self-enforcement of decentralized access control policies. The hierarchical permissioned blockchain architecture enhances scalability, low latency, and high throughput. A proof of concept using Hyperledger Fabric demonstrates the proposed solution's efficiency and effectiveness, and a security analysis is provided, highlighting its ability to ensure availability, integrity, and confidentiality in IoT environments. The sequence diagram of this model is depicted in Figure 6 [55].

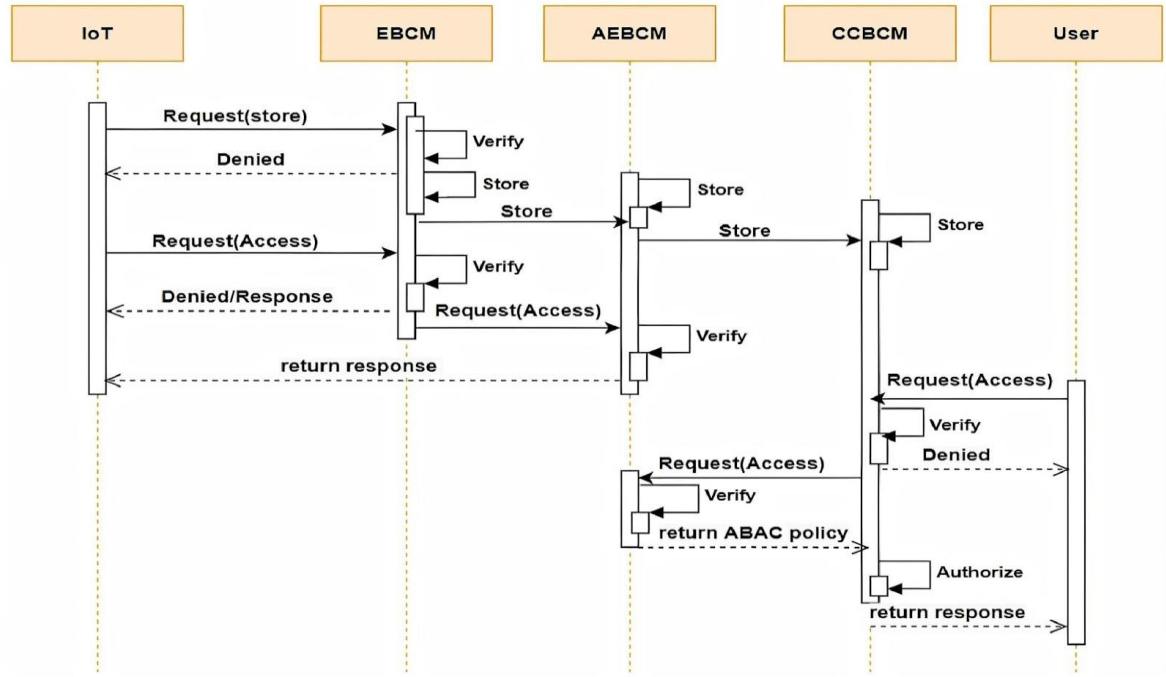

Fig. 6 Sequence diagram of Light-weight Hierarchical Blockchain-based Multi-chaincode Access Control [55]

Access Control Scheme For Implantable Medical Devices: Implantable Medical Devices (IMDs) are critical for diagnostic, monitoring, and therapeutic functions within the human body. However, due to their limited computational power, storage, and battery capacity, IMDs are vulnerable to adversarial attacks, particularly on wireless interfaces. In [56], the authors proposed a novel proxy-based fine-grained access control scheme for IMDs, aiming to extend the IMD's lifetime by offloading heavy cryptographic computations to a proxy device such as a smartphone. The proposed scheme, implemented on real emulator devices, utilizes ciphertext-policy attribute-based encryption (CP-ABE) to enforce fine-grained access control, ensuring only qualified and authorized individuals can access the IMDs. The proxy communicates with the IMD programmer through an audio cable, eliminating the need for patient approval, which is particularly beneficial in emergencies where the patient is unconscious. Additionally, the scheme provides accountability for treatments administered via IMDs in case of medical disputes. Evaluation results demonstrate the lightweight and effective nature of the proposed scheme.

Smart Contract-Based Access Control Scheme (SACS): Saha et al. [57] proposed a new access control model, SACS, to secure medical data exchange in a 6G-enabled healthcare system. The authors argue that current healthcare systems require improved methods to address patient privacy and data security, especially with integrating the Internet of Things and big data analytics. SACS leverages blockchain technology to provide a secure and transparent platform for patients to communicate with healthcare providers and share medical data. SACS offers several advantages, including strong security measures against various attacks and lower computational costs compared to existing solutions.

4. FEATURES OF ACCESS CONTROL MODELS

Achieving robust access control models involves considering numerous features and specifications, particularly those rooted in access control determination factors that assess each model's performance and economic values. These factors serve as essential benchmarks to evaluate the efficacy of various models. The pivotal features in access control research include security, simplicity, flexibility, dynamics, granularity, scalability, and efficiency [47], [48], [58], [59], [60].

Security: This pertains to safeguarding against unauthorized access to sensitive and unrelated information. Access control systems categorize data based on potential damage, leading to the application of tailored measures to preserve privacy and enhance system security. MAC is often regarded as the most secure due to its stringent policy enforcement and centralized

control. ABAC also offers strong security through its fine-grained policies based on multiple attributes. DAC offers flexibility but is more vulnerable to insider threats due to its reliance on users to set permissions. RBAC provides robust security through role hierarchies and separation of duties.

Simplicity: The ease and convenience of implementing and managing an access control system significantly influence model selection. Organizations may opt for a system aligned with their requirements and organizational size, avoiding unnecessary complexity and expenses. DAC is considered the simplest, allowing resource owners to grant or revoke access permissions easily. RBAC, while slightly more complex due to the need to define roles and permissions, still maintains a high level of simplicity compared to MAC. ABAC is the most complex, requiring comprehensive attribute definitions and policy specifications. MAC is also complex, requiring detailed policies set by administrators.

Flexibility: An effective access control system should be able to add, edit, delete, or continuously update data. This adaptability is crucial for responding to changes in security levels, user dynamics, and environmental challenges. Inflexibility can lead to increased administrative workload and unplanned technical costs. ABAC is highly flexible, allowing policies to be based on a wide range of attributes, such as user roles, time of access, and resource type. DAC is also highly flexible, allowing users to change permissions quickly. RBAC offers significant flexibility through its role assignments, enabling easy modification of user roles and permissions as organizational needs evolve. MAC, however, is the least flexible, as it relies on predefined policies that are difficult to alter without comprehensive changes to the security infrastructure.

Dynamicity: The manual assignment of permissions, roles, and policies can burden administrators, necessitating a dynamic approach. Automating these processes streamlines system upgrades and improvements, reducing administrative workload and controlling unexpected costs. ABAC excels in dynamicity, as attribute-based policies can easily adjust to meet changing requirements. RBAC also performs well, with roles that can be quickly adjusted. DAC is dynamic but depends heavily on the users' responsiveness to changing needs. MAC, with its rigid structure, is the least dynamic, often requiring significant administrative effort to accommodate changes.

Granularity: Access control systems implement granularity in two types—fine-grained and coarse-grained. Fine-grained granularity provides precise references to users and resources, while coarse-grained allows broader access control through groups of roles. ABAC and MAC both offer high granularity, with ABAC enabling fine-grained policies based on multiple attributes and MAC allowing detailed, fine-grained policies. RBAC also provides good granularity with detailed role definitions. DAC, while flexible, typically offers coarser granularity, with permissions often set at the file or resource level rather than the more detailed attribute level.

Scalability: The system's ability to adapt to increased users, roles, resources, and policies is vital. A well-designed system should accommodate growth without compromising performance, preventing disruptions and potential financial damage to organizations. ABAC is highly scalable, with its attribute-based approach allowing for flexible and efficient management of large numbers of users and resources. RBAC is also highly scalable, as roles can be easily managed and applied to large numbers of users. MAC scales well in environments with stringent security requirements. DAC, however, can become cumbersome in large-scale environments due to the need for individual permission management.

Efficiency: Evaluating the resources used by the selected access control system is crucial. The model should align with the organization's needs, ensuring effective management and execution of tasks while avoiding unnecessary complexity. Efficiency, in particular, stands out as a critical parameter, serving as a primary indicator among other parameters for a successful access control model. DAC is generally efficient for small to medium-sized environments but can become inefficient as the number of users and permissions grows. RBAC offers a good balance, providing efficient management through role definitions. ABAC can be resource-intensive due to the complexity of its policies, but it provides efficient access control in dynamic environments.

Table 1: Comparison summary of traditional access control models

Criteria	DAC	MAC	RBAC	ABAC
Security	Mid	High	Semi-High	High
Simplicity	High	Mid	High	Low
Flexibility	High	Low	Semi-High	High
Dynamicity	Mid	Low	Semi-High	High
Granularity	Low	High	Mid	High
Scalability	Low	Mid	High	High
Efficiency	Mid	Low	Mid	Mid

MAC, while secure, often incurs higher overhead due to its complex policy enforcement mechanisms.

Table 1 summarizes the comparison of traditional access control models.

5. SECURITY AND VULNERABILITIES

In the OWASP Top 10 Version 2021, the Broken Access Control is ranked first position [61]. Secure access control is a fundamental pillar of software architecture, ensuring only authorized users can access specific resources and functionalities. However, access control models are susceptible to various attacks and vulnerabilities that can be exploited by malicious actors. These security weaknesses can have a significant impact, potentially leading to data breaches, unauthorized modifications, or even complete system compromise. This section explores the common attacks and vulnerabilities associated with access control models. By identifying these weaknesses, we aim to empower software architects and developers to design more robust and secure access control mechanisms, ultimately safeguarding sensitive information and functionalities within software systems.

Privilege Escalation: Attackers constantly seek unauthorized access to resources with higher permissions. This can be achieved through two main methods [62]:

1) Vertical Privilege Escalation: Imagine a standard user account on a computer. A vertical privilege escalation attack aims to exploit vulnerabilities in the system to elevate permissions from a standard user to an administrator, gaining complete control. This can be done by targeting weaknesses in software applications and the operating system or even tricking users into revealing their credentials.

2) Horizontal Privilege Escalation: While vertical escalation grants access to higher levels, horizontal escalation focuses on gaining access to another user's data or functionality at the same privilege level. For instance, an attacker might compromise a user account within a file-sharing system to access and potentially modify documents belonging to colleagues. This can be a stepping stone for further attacks, as compromised user accounts with similar permissions might have access to sensitive information.

Insecure Direct Object References (IDOR): Imagine an online store where users can view their order history. Insecure Direct Object References (IDOR) vulnerabilities arise when applications rely on user-supplied input to control access to resources without proper validation. An attacker could exploit this by crafting requests that manipulate this input to access unauthorized data. In the order history example, an attacker might be able to view someone else's order details by altering the user ID parameter in the URL.

Broken Access Control: This is a broad term encompassing various vulnerabilities that bypass or circumvent access control mechanisms [63]. It can involve weaknesses in authentication (verifying a user's identity), authorization (determining what a user can do), or both. For instance, an attacker might leverage a brute-force attack to crack weak passwords (authentication) or exploit a flaw in the authorization process to access unauthorized functionalities.

Misconfiguration: Access control systems are only as secure as they are configured. Incorrect settings can leave gaping holes for attackers to exploit [64]. This could involve accidentally granting overly broad permissions to user groups or roles or even using weak access control mechanisms altogether. Imagine a database server configured to allow any user with a valid login to access all data. This is a significant misconfiguration that could result in a severe data breach.

Unprotected Functionality: Not all functionalities within a software system are created equal. Some might require stricter access controls than others. Leaving functionalities unprotected, especially those that bypass access control checks entirely, creates significant vulnerabilities [65]. This could be hidden administrative features accessible through undocumented backdoors or APIs (application programming interfaces) that lack proper authentication and authorization.

Parameter-Based Access Control: While convenient, relying solely on parameters within URLs or requests to control access can be risky. These parameters can often be manipulated by attackers. For instance, an e-commerce application might use a parameter in the URL to differentiate between product pages. A malicious actor could potentially modify this parameter to access a hidden administration page or manipulate product information.

Multi-Step Process Vulnerabilities: Some access control mechanisms rely on a sequence of steps to verify user access. The entire process can be compromised if there's a weakness in any single step. Imagine a multi-factor authentication system that sends a one-time code to the user's phone after a successful login attempt. If the system fails to properly validate the one-time code before granting access, the initial login process becomes meaningless.

Fig. 7 Challenges and opportunities in access control field

6. CHALLENGES AND OPEN PROBLEMS

In the rapidly advancing landscape of computer science, access control stands as a pivotal domain, necessitating continuous innovation to meet evolving challenges. As shown in Figure 7, this section explores the current challenges within access control for software architecture, focusing on scalability, adaptability to dynamic environments, fine-grained control, context-aware decision-making, and the integration of emerging technologies like blockchain and artificial intelligence.

As we explore these challenges, we aim to pinpoint existing gaps and pave the way for future advancements. This discussion guides researchers, practitioners, and software architects, offering insights that inform decision-making and inspire new research endeavors. Together, we contribute to the ongoing evolution of access control mechanisms in the dynamic landscape of software architecture.

Scalability: As software systems continue to evolve in complexity and expand in size, the administration of access control policies encounters escalating challenges. The sheer volume of users, roles, and permissions in expansive systems can strain traditional access control models, potentially resulting in performance bottlenecks. Addressing the scalability concern necessitates the development of sophisticated access control models capable of efficiently managing a vast array of users and permissions without compromising system responsiveness. The quest for scalability involves optimizing algorithms, leveraging distributed architectures, and employing innovative data structures to ensure that access control mechanisms remain effective and responsive in the face of system growth.

Dynamic Environments: In the dynamic landscape of modern software ecosystems, where changes in users, roles, and permissions are frequent and often unpredictable, traditional access control models exhibit limitations. These models may struggle to adapt seamlessly to dynamic modifications, potentially leading to disruptions during system updates or modifications. Crafting access control models tailored for dynamic environments is imperative. This involves designing systems that can gracefully accommodate changes without compromising security, ensuring a harmonious

coexistence between the need for system evolution and the imperative to maintain a secure and stable operational environment.

Fine-Grained Access Control: In the pursuit of comprehensive access control, particularly in scenarios where users require nuanced access to diverse facets of the system or its data, the concept of fine-grained access control emerges as pivotal. This entails developing models that transcend traditional role-based structures, allowing for precise and context-aware permission assignment. Beyond roles, these models consider additional factors such as user attributes, environmental conditions, and contextual information. The objective is to enable a more granular and adaptive approach to access permissions, aligning closely with the specific requirements and intricacies of the organizational landscape.

Context-Aware Access Control: Many contemporary systems often lack the capability to assimilate contextual information when making access decisions. In response to this deficiency, research initiatives focus on developing and implementing context-aware access control models. These models integrate factors like temporal considerations, geographical locations, and device-specific characteristics into the decision-making process. By incorporating contextual awareness, access control mechanisms can dynamically adjust permissions based on the situational context, thereby enhancing the adaptability and relevance of access decisions.

Adaptive Security Policies: The prevalence of dynamic and sophisticated security threats necessitates a departure from static access control policies. Traditional models, with predefined and unchanging policies, may prove inadequate in responding effectively to emerging threats. The call to action involves the creation of adaptive access control models that can dynamically evolve their security policies in response to changing threat landscapes. These adaptive models proactively fortify security measures, ensuring a resilient defense against evolving cyber threats through real-time adjustments to access permissions and controls.

Privacy Concerns: While access control mechanisms aim to fortify security, they must concurrently address privacy concerns, especially when dealing with sensitive information. Striking a delicate balance between robust security measures and safeguarding user privacy is a formidable challenge. The design of access control models should incorporate privacy-centric features, ensuring that the mechanisms in place not only defend against unauthorized access but also uphold individual privacy rights. This involves robust encryption, anonymization techniques, and stringent controls on data access to mitigate inadvertent privacy breaches.

Interoperability: In the heterogeneous landscape of modern software environments, e.g., IoT and cloud computing, ensuring the seamless integration and interoperability of diverse access control models poses a critical challenge [13], [66]. The coexistence of disparate platforms, services, and applications demands standardized approaches. Establishing universal standards and protocols becomes paramount, facilitating the integration of various access control mechanisms across the software ecosystem. This interoperability drive aims to create a cohesive security framework that transcends individual components, fostering a unified and effective defense against unauthorized access.

Auditability and Accountability: Tracing and auditing access events are indispensable for accountability, compliance, and forensic analysis. However, achieving effective auditability can be a complex undertaking. Developing mechanisms for comprehensive auditing, logging, and accountability within access control systems involves not only tracking access events but also ensuring the integrity and reliability of the generated audit trail. Research and development efforts must focus on creating robust and standardized methodologies for auditing, enabling organizations to meet regulatory requirements, conduct thorough forensic investigations, and reinforce accountability throughout the system lifecycle.

User-Centric Access Control: Traditional access control models may fall short in empowering end-users to exert control over their access permissions [67]. The evolving landscape of access control advocates for exploring user-centric models that place individuals at the forefront of permission management. Empowering users with the ability to define and manage their access rights fosters a sense of ownership and transparency. User-centric access control models aim to democratize access management, allowing individuals to tailor permissions according to their specific needs, thereby bridging the gap between security imperatives and user autonomy.

Blockchain Integration: Integrating secure and decentralized access control mechanisms, particularly leveraging blockchain technology, represents a frontier in contemporary cybersecurity [55], [67]. Exploring the potential of blockchain-based access control systems introduces a paradigm shift in security, emphasizing transparency, traceability, and decentralization. Blockchain's inherent

characteristics of immutability and distributed consensus offer promising avenues for securing access control transactions. Research endeavors delve into harnessing the power of blockchain to fortify access control, envisioning a future where decentralized ledgers play a pivotal role in enhancing the security fabric of software architectures.

Machine Learning and AI: Leveraging machine learning algorithms to analyze user behavior and automatically adjust access control policies represents a cutting-edge frontier [68]. Incorporating artificial intelligence (AI) techniques enhances the adaptability and responsiveness of access control models. Machine learning algorithms can analyze patterns in user behavior, enabling the automatic adjustment of access control policies based on evolving usage patterns [69]. Moreover, AI techniques contribute to the detection and mitigation of security threats that may compromise access control mechanisms [70]. This intersection of AI and access control holds the potential to fortify security postures and create more resilient defense mechanisms against dynamic cyber threats.

7. CONCLUSION

This paper navigated the diverse landscape of access control in software architecture, exploring models from DAC and MAC to emerging paradigms of blockchain-based access control. We categorized and reviewed these models, considering their usage and functionalities, with a focus on applications in evolving fields like cloud computing and the Internet of Things. Our discussion highlighted existing challenges, including scalability and privacy concerns, underscoring the complexities faced by current models. Looking ahead, we outlined key research directions, including integrating machine learning, addressing scalability, and developing user-centric access control. In the ever-evolving digital landscape, access control remains pivotal for safeguarding organizational assets. This survey aims to contribute to ongoing discussions, inspiring further research and advancements in access control mechanisms within software architecture.

REFERENCES

[1] S. Parkinson and S. Khan, "A Survey on Empirical Security Analysis of Access-control Systems: A Real-world Perspective," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–28, Jul. 2022, doi: 10.1145/3533703.

[2] S. M. Awan, M. A. Azad, J. Arshad, U. Waheed, and T. Sharif, "A Blockchain-Inspired Attribute-Based Zero-Trust Access Control Model for IoT," *Inf.*, vol. 14, no. 2, p. 129, Feb. 2023, doi: 10.3390/info14020129.

[3] R. S. Sandhu and P. Samarati, "Access Control: Principles and Practice," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 40–48, Sep. 1994, doi: 10.1109/35.312842.

[4] J. Moffett, M. Sloman, and K. Twidle, "Specifying discretionary access control policy for distributed systems," *Comput. Commun.*, vol. 13, no. 9, pp. 571–580, Nov. 1990, doi: 10.1016/0140-3664(90)90008-5.

[5] R. S. Sandhu, "Lattice-Based Access Control Models," *Computer (Long. Beach. Calif).*, vol. 26, no. 11, pp. 9–19, Nov. 1993, doi: 10.1109/2.241422.

[6] R. S. Sandhu, "Role-based Access Control," in *Advances in computers*, vol. 46, Elsevier, 1998, pp. 237–286. doi: 10.1016/S0065-2458(08)60206-5.

[7] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Attribute-based access control," *Computer (Long. Beach. Calif).*, vol. 48, no. 2, pp. 85–88, Feb. 2015, doi: 10.1109/MC.2015.33.

[8] M. Zviran and Z. Erlich, "Identification and Authentication: Technology and Implementation Issues," *Commun. Assoc. Inf. Syst.*, vol. 17, no. 1, p. 4, 2006, doi: 10.17705/1cais.01704.

[9] M. Penelova, "Access Control Models," *Cybern. Inf. Technol.*, vol. 21, no. 4, pp. 77–104, Dec. 2021, doi: 10.2478/cait-2021-0044.

[10] J. Park and R. Sandhu, "Towards usage control models: Beyond traditional access control," in *Proceedings of ACM Symposium on Access Control Models and Technologies (SACMAT 2002)*, New York, NY, USA: ACM, Jun. 2002, pp. 57–64. doi: 10.1145/507721.507722.

[11] S. Kirrane, A. Mileo, and S. Decker, "Access control and the Resource Description Framework: A survey," *Semant. Web*, vol. 8, no. 2, pp. 311–352, Dec. 2017, doi: 10.3233/SW-160236.

[12] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of internet of things," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4682–4696, Jun. 2020, doi: 10.1109/JIOT.2020.2969326.

[13] F. Cai, N. Zhu, J. He, P. Mu, W. Li, and Y. Yu, "Survey of access control models and technologies for cloud computing," *Cluster Comput.*, vol. 22, no. S3, pp. 6111–6122, May 2019, doi: 10.1007/s10586-018-1850-7.

[14] M. U. Aftab *et al.*, "Traditional and Hybrid Access Control Models: A Detailed Survey," *Secur. Commun. Networks*, vol. 2022, pp. 1–12, Feb. 2022, doi: 10.1155/2022/1560885.

[15] M. U. Aftab *et al.*, "A Hybrid Access Control Model with Dynamic COI for Secure Localization of Satellite and IoT-Based Vehicles," *IEEE Access*, vol. 8, pp. 24196–24208, 2020, doi: 10.1109/ACCESS.2020.2969715.

[16] M. Sookhak, F. R. Yu, M. K. Khan, Y. Xiang, and R. Buyya, "Attribute-based data access control in mobile cloud computing: Taxonomy and open issues," *Futur. Gener. Comput. Syst.*, vol. 72, pp. 273–287, Jul. 2017, doi: 10.1016/j.future.2016.08.018.

[17] V. C. Hu *et al.*, "Guide to attribute based access

control (abac) definition and considerations," Citeseer, Gaithersburg, MD, Jan. 2014. doi: 10.6028/NIST.SP.800-162.

[18] D. F. Ferraiolo, R. Chandramouli, V. C. Hu, and D. R. R. Kuhn, "A Comparison of Attribute Based Access Control (ABAC) Standards for Data Service Applications," Gaithersburg, MD, Oct. 2016. doi: 10.6028/NIST.SP.800-178.

[19] Q. M. Rajpoot, C. D. Jensen, and R. Krishnan, "Integrating attributes into role-based access control," in *Data and Applications Security and Privacy XXIX: 29th Annual IFIP WG 11.3 Working Conference, DBSec 2015, Fairfax, VA, USA, July 13-15, 2015, Proceedings 29*, Springer, 2015, pp. 242–249. doi: 10.1007/978-3-319-20810-7_17.

[20] Q. M. Rajpoot, C. D. Jensen, and R. Krishnan, "Attributes enhanced role-based access control model," in *Trust, Privacy and Security in Digital Business: 12th International Conference, TrustBus 2015, Valencia, Spain, September 1-2, 2015, Proceedings 12*, Springer, 2015, pp. 3–17. doi: 10.1007/978-3-319-22906-5_1.

[21] Y. Xu, W. Gao, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "A Feasible Fuzzy-Extended Attribute-Based Access Control Technique," *Secur. Commun. Networks*, vol. 2018, pp. 1–11, Jun. 2018, doi: 10.1155/2018/6476315.

[22] B. Jiang, Q. He, M. He, Z. Zhai, and B. Zhao, "FACSC: Fine-Grained Access Control Based on Smart Contract for Terminals in Software-Defined Network," *Secur. Commun. Networks*, vol. 2023, pp. 1–13, May 2023, doi: 10.1155/2023/6013270.

[23] X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering DAC, MAC and RBAC," in *Data and Applications Security and Privacy XXVI: 26th Annual IFIP WG 11.3 Conference, DBSec 2012, Paris, France, July 11-13, 2012. Proceedings 26*, Springer, 2012, pp. 41–55. doi: 10.1007/978-3-642-31540-4_4.

[24] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 224–274, Aug. 2001, doi: 10.1145/501978.501980.

[25] J. S. Park, R. Sandhu, and G. J. Ahn, "Role-Based Access Control on the Web," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 1, pp. 37–71, Feb. 2001, doi: 10.1145/383775.383777.

[26] B. Carminati, E. Ferrari, and A. Perego, "Rule-based access control for social networks," in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops: OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SeB*, Springer, 2006, pp. 1734–1744. doi: 10.1007/11915072_80.

[27] M. A. Al-Kahtani and R. Sandhu, "Induced role hierarchies with attribute-based RBAC," in *Proceedings of the eighth ACM symposium on Access control models and technologies*, New York, NY, USA: ACM, Jun. 2003, pp. 142–148. doi: 10.1145/775412.775430.

[28] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: a temporal role-based access control model," in *Proceedings of the fifth ACM workshop on Role-based access control*, New York, NY, USA: ACM, Jul. 2000, pp. 21–30. doi: 10.1145/344287.344298.

[29] E. Uzun *et al.*, "Analyzing temporal role based access control models," in *Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT*, New York, NY, USA: ACM, Jun. 2012, pp. 177–186. doi: 10.1145/2295136.2295169.

[30] V. Takalkar and P. N. Mahalle, "Trust-Based Access Control in Multi-role Environment of Online Social Networks," *Wirel. Pers. Commun.*, vol. 100, no. 2, pp. 391–399, May 2018, doi: 10.1007/s11277-017-5078-2.

[31] B. Gwak, J. H. Cho, D. Lee, and H. Son, "TARAS: Trust-Aware Role-Based Access Control System in Public Internet-of-Things," in *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, IEEE, Aug. 2018, pp. 74–85. doi: 10.1109/TrustCom/BigDataSE.2018.00022.

[32] S. Vahabli and R. Ravanmehr, "A novel trust-based access control for social networks using fuzzy systems," *World Wide Web*, vol. 22, no. 6, pp. 2241–2265, Nov. 2019, doi: 10.1007/s11280-019-00668-y.

[33] I. Ray, M. Kumar, and L. Yu, "LRBAC: A location-aware role-based access control model," in *Information Systems Security: Second International Conference, ICISS 2006, Kolkata, India, December 19-21, 2006. Proceedings 2*, Springer, 2006, pp. 147–161. doi: 10.1007/11961635_10.

[34] M. Uddin, S. Islam, and A. Al-Nemrat, "A Dynamic Access Control Model Using Authorising Workflow and Task-Role-Based Access Control," *IEEE Access*, vol. 7, pp. 166676–166689, 2019, doi: 10.1109/ACCESS.2019.2947377.

[35] N. Solanki, Y. Huang, I. L. Yen, F. Bastani, and Y. Zhang, "Resource and Role Hierarchy Based Access Control for Resourceful Systems," in *Proceedings - International Computer Software and Applications Conference*, IEEE, Jul. 2018, pp. 480–486. doi: 10.1109/COMPSAC.2018.10280.

[36] T. Y. Lin, "Managing information flows on discretionary access control models," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, IEEE, Oct. 2006, pp. 4759–4762. doi: 10.1109/ICSMC.2006.385057.

[37] S. Osborn, R. Sandhu, and Q. Munawer, "Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 2, pp. 85–106, May 2000, doi: 10.1145/354876.354878.

[38] R. Kumar and R. Tripathi, "Scalable and secure

access control policy for healthcare system using blockchain and enhanced Bell–LaPadula model," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 2, pp. 2321–2338, Feb. 2021, doi: 10.1007/s12652-020-02346-8.

[39] L. Guo, X. Yang, and W. C. Yau, "TABE-DAC: Efficient Traceable Attribute-Based Encryption Scheme with Dynamic Access Control Based on Blockchain," *IEEE Access*, vol. 9, pp. 8479–8490, 2021, doi: 10.1109/ACCESS.2021.3049549.

[40] S. Khare and A. Badholia, "BLA2C2: Design of a Novel Blockchain-based Light-Weight Authentication & Access Control Layer for Cloud Deployments," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 3, pp. 283–294, Apr. 2023, doi: 10.17762/ijritcc.v11i3.6359.

[41] M. A. Madani, A. Kerkri, and M. Aissaoui, "MC-ABAC: An ABAC-based Model for Collaboration in Multi-Cloud Environment," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 6, pp. 1182–1190, 2023, doi: 10.14569/IJACSA.2023.01406126.

[42] J. Luo, H. Wang, X. Gong, and T. Li, "A Novel Role-based Access Control Model in Cloud Environments," *Int. J. Comput. Intell. Syst.*, vol. 9, no. 1, pp. 1–9, 2016, doi: 10.1080/18756891.2016.1144149.

[43] C. Uikey and D. S. Bhilare, "RBACA: Role-based access control architecture for multi-domain cloud environment," *Int. J. Bus. Inf. Syst.*, vol. 28, no. 1, pp. 1–17, 2018, doi: 10.1504/IJBIS.2018.091160.

[44] A. Singh, A. Kumar, and S. Namasudra, "DNACDS: Cloud IoE big data security and accessing scheme based on DNA cryptography," *Front. Comput. Sci.*, vol. 18, no. 1, p. 181801, Feb. 2024, doi: 10.1007/s11704-022-2193-3.

[45] J. Guo, C. Tian, X. Lu, L. Zhao, and Z. Duan, "Multi-keyword ranked search with access control for multiple data owners in the cloud," *J. Inf. Secur. Appl.*, vol. 82, p. 103742, May 2024, doi: 10.1016/j.jisa.2024.103742.

[46] C. Daudén-Esmel, J. Castellà-Roca, and A. Viejo, "Blockchain-based access control system for efficient and GDPR-compliant personal data management," *Comput. Commun.*, vol. 214, pp. 67–87, Jan. 2024, doi: 10.1016/j.comcom.2023.11.017.

[47] A. Thakare, E. Lee, A. Kumar, V. B. Nikam, and Y. G. Kim, "PARBAC: Priority-Attribute-Based RBAC Model for Azure IoT Cloud," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2890–2900, Apr. 2020, doi: 10.1109/JIOT.2019.2963794.

[48] M. Alam, N. Emmanuel, T. Khan, Y. Xiang, and H. Hassan, "Garbled role-based access control in the cloud," *J. Ambient Intell. Humaniz. Comput.*, vol. 9, no. 4, pp. 1153–1166, Aug. 2018, doi: 10.1007/s12652-017-0573-6.

[49] R. Zhang, G. Liu, S. Li, Y. Wei, and Q. Wang, "ABSAC: Attribute-based access control model supporting anonymous access for smart cities," *Secur. Commun. Networks*, vol. 2021, pp. 1–11, Mar. 2021, doi: 10.1155/2021/5531369.

[50] N. Kaaniche and M. Laurent, "Attribute-based signatures for supporting anonymous certification," in *Computer Security–ESORICS 2016: 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part I 21*, Springer, 2016, pp. 279–300. doi: 10.1007/978-3-319-45744-4_14.

[51] B. Wang, W. Li, and N. N. Xiong, "Time-Based Access Control for Multi-attribute Data in Internet of Things," *Mob. Networks Appl.*, vol. 26, no. 2, pp. 797–807, Apr. 2021, doi: 10.1007/s11036-019-01327-2.

[52] M. U. Aftab *et al.*, "Negative Authorization by Implementing Negative Attributes in Attribute-Based Access Control Model for Internet of Medical Things," in *Proceedings - 15th International Conference on Semantics, Knowledge and Grids: On Big Data, AI and Future Interconnection Environment, SKG 2019*, IEEE, Sep. 2019, pp. 167–174. doi: 10.1109/SKG49510.2019.00036.

[53] S. F. Aghili, M. Sedaghat, D. Singelée, and M. Gupta, "MLS-ABAC: Efficient Multi-Level Security Attribute-Based Access Control scheme," *Futur. Gener. Comput. Syst.*, vol. 131, pp. 75–90, Jun. 2022, doi: 10.1016/j.future.2022.01.003.

[54] V. Karnatak, A. K. Mishra, N. Tripathi, M. Wazid, J. Singh, and A. K. Das, "A secure signature-based access control and key management scheme for fog computing-based IoT-enabled big data applications," *Secur. Priv.*, vol. 7, no. 2, p. e353, Mar. 2024, doi: 10.1002/spy2.353.

[55] A. I. Abdi *et al.*, "Hierarchical Blockchain-Based Multi-Chaincode Access Control for Securing IoT Systems," *Electron.*, vol. 11, no. 5, p. 711, Feb. 2022, doi: 10.3390/electronics11050711.

[56] L. Wu and J. Du, "Designing novel proxy-based access control scheme for implantable medical devices," *Comput. Stand. Interfaces*, vol. 87, p. 103754, Jan. 2024, doi: 10.1016/j.csi.2023.103754.

[57] S. Saha, A. K. Das, M. Wazid, Y. Park, S. Garg, and M. Alrashoud, "Smart Contract-Based Access Control Scheme for Blockchain Assisted 6G-Enabled IoT-Based Big Data Driven Healthcare Cyber Physical Systems," *IEEE Trans. Consum. Electron.*, pp. 1–1, 2024, doi: 10.1109/TCE.2024.3391667.

[58] S. Long and L. Yan, "RACAC: An Approach toward RBAC and ABAC Combining Access Control," in *2019 IEEE 5th International Conference on Computer and Communications, ICCC 2019*, IEEE, Dec. 2019, pp. 1609–1616. doi: 10.1109/ICCC47050.2019.9064301.

[59] M. U. Aftab *et al.*, "Permission-Based Separation of Duty in Dynamic Role-Based Access Control Model," *Symmetry (Basel).*, vol. 11, no. 5, p. 669, May 2019, doi: 10.3390/sym11050669.

[60] J. Huang, D. M. Nicol, R. Bobba, and J. H. Huh, "A framework integrating attribute-based policies into role-based access control," in *Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT*, New York, NY, USA: ACM, Jun. 2012, pp. 187–196. doi: 10.1145/2295136.2295170.

[61] "OWASP Top Ten." Accessed: Jun. 26, 2024. [Online]. Available: https://owasp.org/www-project-top-ten/

[62] M. Mehmood, R. Amin, M. M. A. Muslam, J. Xie, and H. Aldabbas, "Privilege Escalation Attack Detection and Mitigation in Cloud Using Machine Learning," *IEEE Access*, vol. 11, pp. 46561–46576, 2023, doi: 10.1109/ACCESS.2023.3273895.

[63] E. Almushiti, R. Zaki, N. Thamer, and R. Alshaya, "An Investigation of Broken Access Control Types, Vulnerabilities, Protection, and Security," in *International Conference on Innovation of Emerging Information and Communication Technology*, Springer, 2023, pp. 253–269. doi: 10.1007/978-3-031-53237-5_16.

[64] T. Xu, L. Jin, X. Fan, Y. Zhou, S. Pasupathy, and R. Talwadker, "Hey, you have given me too many knobs!: Understanding and dealing with over-designed configuration in system software," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, 2015, pp. 307–319. doi: 10.1145/2786805.2786852.

[65] T. Xu and Y. Zhou, "Systems approaches to tackling configuration errors: A survey," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 1–41, 2015, doi: 10.1145/2791577.

[66] E. Bertin, D. Hussein, C. Sengul, and V. Frey, "Access control in the Internet of Things: a survey of existing approaches and open research questions," *Ann. des Telecommun. Telecommun.*, vol. 74, no. 7–8, pp. 375–388, Aug. 2019, doi: 10.1007/s12243-019-00709-7.

[67] S. H. Hashemi, F. Faghri, and R. H. Campbell, "Decentralized User-Centric Access Control using PubSub over Blockchain," *arXiv Prepr. arXiv1710.00110*, Sep. 2017, [Online]. Available: http://arxiv.org/abs/1710.00110

[68] K. Istiaque Ahmed, M. Tahir, M. Hadi Habaebi, S. Lun Lau, and A. Ahad, "Machine Learning for Authentication and Authorization in IoT: Taxonomy, Challenges and Future Research Direction," *Sensors*, vol. 21, no. 15, p. 5122, Jul. 2021, doi: 10.3390/s21155122.

[69] S. Aboukadri, A. Ouaddah, and A. Mezrioui, "Machine learning in identity and access management systems: Survey and deep dive," *Comput. Secur.*, vol. 139, p. 103729, Apr. 2024, doi: 10.1016/j.cose.2024.103729.

[70] L. Zhang *et al.*, "ACFIX: Guiding LLMs with Mined Common RBAC Practices for Context-Aware Repair of Access Control Vulnerabilities in Smart Contracts," *arXiv Prepr. arXiv2403.06838*, Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.06838

مراجعة الأدبيات حول نماذج التحكم في الوصول في معمارية البرمجيات

علي صحفي sohofi@shdu.ac.ir

امير دهباشي دزفولي amir.dehbashi@shdu.ac.ir

قسم هندسة الحاسوب, جامعة شهاب دانيش, قم, ايران

تاريخ الاستلام: ا مايو 2024 استلم بصيغته المنقحة: 1 يوليو 2024 تاريخ القبول: 28 يوليو 2024

الملخص

في مجال هندسة البرمجيات، يعد ضمان أمن الأصول التنظيمية ضد الوصول غير المصرح به أمرًا بالغ الأهمية للأمن السيبراني. تجري هذه الورقة استكشافًا شاملاً للتحكم في الوصول، بدءًا من النماذج التقليدية مثل DAC *و*MAC *و*RBAC *و*ABAC *إلى الاتجاهات المعاصرة مثل التحكم في الوصول القائم على السياسات والتحكم في الوصول القائم على* blockchain*. تقوم بتصنيف نماذج التحكم في الوصول بناءً على استخدامها مع الأخذ في الاعتبار المجالات الناشئة الجديدة مثل الحوسبة السحابية وإنترنت الأشياء. وبعد ذلك، نتعمق في التحديات الحالية ونحدد اتجاهات البحث المستقبلية.*

الكلمات الدالة :

صلاحية التحكم صلاحية الدخول، نماذج التحكم في الوصول، نهج الوصول، معمارية البرمجيات، الحماية.