УДК 004.056

N. R. Hryhoruk

ANALYSIS OF ACCESS CONTROL MODELS

Vinnytsia National Technical University

Анотація

У роботі проаналізовано моделі контролю доступу для багаторівневих інформаційних систем: дискреційну (DAC), мандатну (MAC), рольову (RBAC) та на основі атрибутів (ABAC). Визначено переваги та недоліки кожної з моделей. За результатами аналізу визначено перспективність їх застосування в різних бізнес-процесах різних організацій.

Ключові слова: управління доступом, кібербезпека, порівняльний аналіз, моделі розмежування прав доступу, застосування.

Abstract

The paper analyzes access control models for multilevel information systems: discretionary (DAC), mandatory (MAC), role-based (RBAC), and attribute-based (ABAC). The advantages and disadvantages of each model are defined. Their application prospects for different organizations' business processes are determined.

Keywords: access management, cybersecurity, comparative analysis, access control models, application.

Introduction

Access control is one of the most important security mechanisms protecting information systems from unauthorized access. Today's cyber threats show that poor access control settings are responsible for many data breaches. A key example is Australia's My Health Record (MHR) system, where access control failures caused incidents to rise from 35 in 2018 to 42 in 2019 [1]. Though officially labeled "administrative errors," these security lapses led many citizens to opt out of the system due to privacy concerns. Identity theft through compromised authentication systems represents another access control failure. Research examining SMS-based two-factor authentication found that 76.5% of banking authentication messages exposed vulnerabilities that could compromise customer accounts [1]. The Westpac Bank incident further underlines this problem, where hackers exploited the PayID lookup service to expose the private details of almost 100,000 customers, information that could enable future fraud [1]. These examples illustrate importance of proper architectural and configuring decisions for identity and access management systems. The latter makes access control models comparative analysis topical.

The objective of this work is to improve the confidentiality protection through comparative analysis of access control models and their application limitations defining.

Research tasks:

1. Analyze the most wide-spread access control models (DAC, MAC, RBAC, ABAC) and their implementation mechanisms.
2. Identify advantages, disadvantages, and use-cases for each model.
3. Defining quality metrics for the models.
4. Develop practical recommendations for selecting and combining access control models in order to achieve multilevel security.

Research results

Discretionary Access Control (DAC) is the most common access control model, used in Linux and Windows through Access Control Lists (ACLs) [2]. In this model, resource owners decide who can access their files and what permissions are to be granted. This makes DAC very flexible and easy to implement without needing centralized security administration [2]. DAC requires minimal infrastructure most operating systems include it as a built-in feature. However, DAC has serious security issues. When malware gets on someone's computer, it automatically gets all their access rights and can spread to other files. Users may accidentally grant the wrong people access to important files. There's no way to control access across the whole company because each person manages their own permissions. Confidentiality protection level cannot be guaranteed while using DAC due to users are able to inadvertently or maliciously grant excessive permissions. It doesn't scale well because permission management becomes chaotic as the organization grows. DAC is fundamentally unsuitable for multilevel security because users can modify permissions in ways that violate multilevel policies. This makes DAC unsuitable for organizations handling confidential information or systems with multiple security levels.

Mandatory Access Control takes the opposite approach from DAC. Instead of users controlling access, security administrators create strict rules that nobody can change. The system assigns security labels to files and users (like "Top Secret", "Secret", "Confidential"), and these labels determine access rights [2]. MAC uses formal models like Bell-LaPadula and Biba. Bell-LaPadula prevents information leaks through "no read up, no write down" rules (users can't read higher classified data or write to lower levels). Biba protects data integrity with "no read down, no write up" rules, but is impractical from the confidentiality protection standpoint. A practical MAC implementation is Security-Enhanced Linux (SELinux), developed by the NSA, which provides an additional access control layer on top of standard Linux DAC. SELinux works through Linux Security Modules (LSM), a kernel subsystem that intercepts system calls before execution, giving administrators full control over what the system allows based solely on defined policy rules [4].

MAC provides strong security guarantees through mathematically proven security models. Military and intelligence agencies rely on MAC specifically because it offers formal assurance that classified information cannot leak to unauthorized parties. Even if hackers compromise someone's account, they can't access information above that person's clearance level. The system enforces rules automatically, so there's no way for users to accidentally leak classified information. MAC was specifically designed for multilevel security requirements and remains the standard for classified information systems. However, the model introduces substantial complexity in policy administration. Setting up all the security levels and rules takes a lot of work, and the system is inflexible   it's hard to share information between departments even when there's a good reason. Despite these problems, MAC is necessary for military systems, intelligence agencies, and any organization handling classified information where security matters more than convenience [2].

Role-Based Access Control (RBAC) creates an organizational layer between users and permissions by assigning access rights to roles instead of individual people [1]. For instance, all employees with the "Accountant" role automatically get permissions to use accounting software, access financial reports, and edit budget spreadsheets. When someone changes jobs, administrators just change their role assignment instead of manually adjusting dozens of individual permissions. This approach offers major benefits for medium and large organizations. RBAC reduces administrative work significantly because permissions are managed at the role level rather than for each person separately. A company with 500 employees might need only 20-30 roles instead of managing 500 individual permission sets as it would be needed in case of the DAC utilization. It also makes the security setup easier to understand and audit since roles typically match job functions that everyone recognizes. RBAC supports principle of least privilege by ensuring users only get permissions necessary for their job functions. Companies using ERP or CRM systems often implement RBAC because it fits naturally with organizational structure [5]. The downside is that complex organizations often end up with too many roles. If accountants in different departments need slightly different permissions, IT might create separate roles for each group until there are hundreds of roles to manage [1]. This "role explosion" defeats the whole purpose. Also, RBAC can't handle rules like "only allow access during business hours" or "block access from foreign countries" because it doesn't consider context [2]. Still, RBAC works well in most corporate environments where organizational structure is stable.

Attribute-Based Access Control (ABAC) is the most flexible and sophisticated access control model available today. Instead of making decisions based only on user’s identity, ABAC evaluates multiple attributes about the user, the resource being accessed, and the current situation [3]. This fine-grained approach gives ABAC exceptional power and adaptability. Organizations can write complex policies that consider location, time, device security status, risk level, and many other factors without creating thousands of roles. ABAC eliminates the role explosion problem by expressing authorization logic directly through attribute relationships. The model supports dynamic access decisions that respond to changing contexts   for example, automatically blocking access when threat levels increase or when users travel to high-risk countries. ABAC works especially well for cloud systems and distributed networks where users, devices, and resources are spread across different locations and security contexts change constantly.

However, ABAC comes with significant challenges. Implementing it requires substantial technical infrastructure to collect and verify attributes from various sources. Organizations need attribute repositories, policy decision points, policy enforcement points, and mechanisms to keep attributes up-to-date. Policies become complicated to write, test, and debug because they involve complex logical expressions [3]. A single policy might combine ten or more attribute conditions with AND, OR, and NOT operators, making it difficult to predict policy behavior. Every access decision requires the system to check multiple attributes in real-time, which can slow down performance. Each attribute query adds latency, and complex policies might require dozens of queries. Organizations also need robust systems for managing attribute accuracy and trust incorrect attributes lead to incorrect access decisions. These difficulties mean ABAC is mainly used by large enterprises and cloud providers who need its advanced capabilities and can afford the implementation investment [3].

Some specialized models handle specific situations beyond the four main access control types. Capability-Based Access Control [6] uses digital tokens or keys that grant specific permissions to holders, working particularly well in distributed systems where traditional identity verification may be challenging. These capabilities can be passed between users and systems, enabling flexible delegation of access rights. The Chinese Wall Policy, also known as the Brewer-Nash model, prevents conflicts of interest by dynamically restricting access based on prior information exposure – such as stopping a consultant from accessing confidential data of competing companies after working with one of them [6]. Risk-Based Access Control (RiskBAC) represents an adaptive approach that continuously adjusts permissions based on real-time risk assessments, considering factors like current threat levels, unusual user behavior patterns, suspicious login locations, or anomalous access requests. This model can automatically restrict or enhance access privileges as risk conditions change [6]. While these specialized models address important security requirements in specific contexts, they are not as widely adopted as DAC, MAC, RBAC, and ABAC in general-purpose information systems.

Table 1 presents a comprehensive comparison of the main access control models based on key security and operational criteria.

Table 1. Comparative characteristics of access control models

Criterion	DAC	MAC	RBAC	ABAC
Flexibility	High	Very Low	Medium	High
Administrative complexity	High	Low	Medium	High
Scalability	Low	Medium	Medium	Medium
Context awareness	No	No	Limited	Yes
Role explosion risk	N/A	N/A	High	Low

Based on the comparative analysis, the following practical recommendations can be formulated:

1. For small organizations with low security requirements: DAC is suitable due to its simplicity and minimal cost, but only when handling non-sensitive data.
2. For military and government systems: MAC is essential when handling classified information requiring formal security guarantees and strict multilevel protection.
3. For corporate environments: RBAC provides the optimal balance between security and usability, particularly effective in organizations with stable hierarchical structures.
4. For dynamic cloud environments: ABAC is recommended when context-aware, fine-grained access control is required, despite higher implementation costs.
5. Hybrid approaches: Many organizations benefit from combining models – for example, using RBAC for general operations while applying MAC to classified data, or layering ABAC policies over RBAC foundations.

All traditional models (DAC, MAC, RBAC) share a fundamental weakness – they are static and cannot adapt to dynamic contexts and real-time risk changes. This limitation becomes critical in modern distributed systems where security threats evolve rapidly and access decisions need to consider environmental factors, user behavior patterns, and current threat levels. This gap is partially addressed by ABAC and specialized models like Risk-Based Access Control, but at the cost of significantly increased complexity and implementation overhead.

Conclusions

Real-world incidents like the Australian My Health Record breaches and Westpac exploitation demonstrate that access control failures remain critical vulnerabilities. This analysis covers four primary access control models for multilevel information systems: DAC, MAC, RBAC, and ABAC. MAC provides the strongest security guarantees, making it essential for military and government systems. RBAC balances security and usability for corporate environments. ABAC offers superior flexibility through context-aware policies, ideal for dynamic cloud systems. DAC provides weak security unsuitable for multilevel environments. Organizations should select models based on security requirements, with hybrid approaches such as combining

RBAC with MAC often proving most effective. Future research should focus on lightweight ABAC implementations and AI-driven policy automation.

REFERENCES

1.Kayes A. S. M., Rahayu W., Dillon T., Chang E., Han J. A Survey of Context-Aware Access Control Mechanisms for Cloud and Fog Networks: Taxonomy and Open Research Issues. Sensors. 2020. Vol. 20, No 9. Article 2464. P. 1–37. DOI: 10.3390/s20092464.

2.Mohamed A. K. Y. S., Auer D., Hofer D., Küng J. A systematic literature review for authorization and access control: definitions, strategies and models. International Journal of Web Information Systems. 2022. Vol. 18, No 2-3. P. 156–180. DOI: 10.1108/IJWIS-04-2022-0077.

3.Hu V. C., Ferraiolo D., Kuhn R., Schnitzer A., Sandlin K., Miller R., Scarfone K. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication 800-162. Gaithersburg, MD : National Institute of Standards and Technology, 2014 (updated 2019). 47 p. DOI: 10.6028/NIST.SP.800-162.

4.What is SELinux? *Red Hat* URL: https://www.redhat.com/en/topics/linux/what-is-selinux (Last accessed: 16.10.2025).

5.Sandhu R. S., Coyne E. J., Feinstein H. L., Youman C. E. Role-Based Access Control Models. IEEE Computer. 1996. Vol. 29, No 2. P. 38–47. DOI: 10.1109/2.485845.

6.Atlam, H.F., Azad, M.A., Alassafi, M.O., Alshdadi, A.A., & Alenezi, A. (2020). "Risk-Based Access Control Model: A Systematic Literature Review." *Future Internet*, 12(6), 103. DOI: 10.3390/fi12060103

Григорук Надія Романівна – студентка групи 2БС-22б, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: grigoruknadia15@gmail.com

Науковий керівник: ***Баришев Юрій Володимирович*** – к.т.н., доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця, email: yuriy.baryshev@vntu.edu.ua

Nadiia Hryhoruk – student of group 2BS-22b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: grigoruknadia15@gmail.com

Scientific supervisor: ***Yurii Baryshev*** – PhD (Eng), Associated Professor of Information Protection Department, Vinnytsia National Technical University, Vinnytsia, Ukraine, yuriy.baryshev@vntu.edu.ua