

RAFIA: A Game-theoretic Risk-based Framework for Insider Threat Mitigation

Hala Yousif Mohamed Ahmed*

College of Computer Science and Information Technology, Sudan University of Science and Technology, Khartoum, 11111, Sudan

E-mail: halaahmed043@hotmail.com

ORCID iD: <https://orcid.org/0000-0002-4727-0337>

*Corresponding Author

Mohamed Mejri

Department of Computer Science and Software Engineering, Université Laval, Quebec, G1V 0A6, Canada

E-mail: mohamed.mejri@ift.ulaval.ca

ORCID iD: <https://orcid.org/0000-0003-4820-3176>

Received: June 5, 2026; Revised: June 24, 2026; Accepted: July 17, 2026; Published: August 8, 2026

Abstract: Insider attacks pose a significant security threat precisely because they originate from individuals with authorized access, making them inherently difficult to detect and prevent; addressing this issue is crucial for preserving the confidentiality, integrity, and availability of organizational systems. This paper contributes to mitigating insider attacks by proposing an approach called RAFIA, which monitors the system, evaluates the risk of insider threats, and blocks malicious actions before unauthorized or high-risk access is granted. The security policy is specified using an enhanced version of Linear Temporal Logic, called Risk-LTL, which evaluates the risk of each new action based on system history, including log files or traces, and a risk evaluation function provided as input. Risk evaluation is based on combining maliciousness probability and impact assessment, enabling quantitative estimation of the risk associated with user actions and action traces. Access decisions are governed by configurable risk thresholds specified within Risk-LTL policies. To strengthen decision-making, the model frames access control as a game between users and the organization. By applying game-theoretic tools, the system analyzes user behavior and makes access decisions that discourage malicious actions and reward honest ones. The objective is to reach a *Nash equilibrium*, where both players act rationally and securely. The proposed approach aims to improve the effectiveness of access control by reducing dishonest behavior and promoting more stable, risk-aware system interactions. Experimental evaluation using synthetic workloads of up to 10,000 access requests demonstrated the practicality of the proposed framework. RAFIA achieved an average authorization latency of approximately 3.3 ms while improving the F1-score compared with a conventional static-threshold access-control baseline.

Index Terms: Access control, Insider threats, Risk-based access control, Game-theoretic access control, Linear Temporal Logic, Risk-LTL

1. Introduction

Over the past few decades, the widespread use of digital technologies has brought growing concerns about data security and privacy [1,2]. Companies today manage a wide range of sensitive information—such as customer records, internal strategies, and intellectual property—that, if exposed, could lead to serious financial and reputational consequences [2]. In recent years, the percentage of organizations reporting insider threats has increased from less than half in 2021 to more than three-quarters by 2024, as shown in Fig. 1 [3]. Recent industry reports further indicate that insider threats continue to grow in both frequency and complexity, driven by the increasing adoption of generative AI, cloud services, and remote work environments. More than half of surveyed organizations reported an increase in insider incidents during the past year, while nearly two-thirds now consider insider threats to be a greater concern than external attacks [4]. These threats originate from individuals within the organization—employees, contractors, or partners—who already have authorized access, making them difficult to detect and prevent. Moreover, costs continue to rise. The Ponemon Institute reports that expenses related to insider threats increased from \$6.6 million in 2022 to \$7.2 million in

2023 [5]. Accordingly, many companies have faced internal security incidents, and in some cases, these incidents occurred multiple times. The main reasons appear to be weak controls over who can access what, insufficient monitoring, and the increasing number of remote workers [6,7]. All these factors increase the likelihood that insiders may engage in malicious or unauthorized activities.

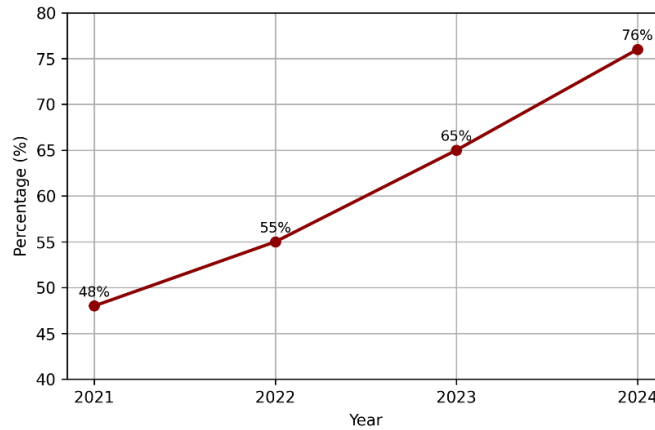


Fig. 1. Percentage of organizations reporting insider threats between 2021 and 2024.

Because of that, companies need to strengthen their security measures and develop more effective mechanisms for the early detection of insider threats.

One way to address these problems is by creating an efficient access-control system. The primary objective of access control is to prevent unauthorized users from gaining access and to limit the permissions of authorized users on protected resources. Additionally, it aims to prevent actions that could lead to security breaches [8]. Access control models can generally be categorized into two types: traditional and dynamic.

Traditional access-control systems rely on fixed, pre-defined policies to decide whether access should be granted. Because these policies are static, they apply the same rules regardless of varying conditions, which limits their effectiveness in adapting to change. As a result, such inflexible methods struggle to provide strong security in highly dynamic and distributed environments [9]. In contrast, dynamic access-control models incorporate not only static rules but also real-time and contextual information—such as user trust levels, historical behavior, location, time, and risk assessments—to make more flexible and context-aware authorization decisions [10]. At the core of these models lies the idea that access decisions are influenced not only by predefined policies but also by real-time contextual factors evaluated at the moment the request is made [11]. This approach enhances flexibility and allows the system to respond more effectively to varying circumstances and situational demands during access control, which can make it more suitable for dynamic and distributed environments than traditional models.

Among the dynamic models, the risk-based access control model evaluates the security risk associated with each access request in real time to inform the access decision.

The risk-based access control model is a dynamic approach that evaluates the security risk associated with each access request to determine authorization decisions [12]. Compared to traditional static models, the risk-based access control model enhances flexibility and robustness by incorporating contextual and real-time information when managing access to system resources. This capability is particularly important in environments that experience irregular or unforeseen access requests, such as healthcare and military applications, where granting timely access can be critical for safety and operational success [13]. Ultimately, the primary objective of the risk-based access control model is to foster secure information sharing that maximizes organizational value while ensuring user accountability and minimizing the likelihood of damage resulting from the unauthorized disclosure of sensitive information [14].

This paper proposes RAFIA, a game-theoretic risk-based framework for the automatic detection and prevention of insider attacks. To achieve this objective, the paper addresses the following research questions:

- **RQ1:** How can insider attacks be specified?
- **RQ2:** How can the risk associated with an insider attack be estimated?
- **RQ3:** How can insider attacks be detected and prevented?

The contributions of this paper are:

- **C1:** A practical model for estimating the risk associated with insider attacks.
- **C2:** A temporal logic, called Risk-LTL, for specifying insider attacks based on Linear Temporal Logic (LTL) and risk evaluation.

- **C3:** An approach, called RAFIA, which takes as input a security policy in Risk-LTL, a risk-evaluation function and continuously monitors activities inside an information system to block any action that makes the risk of an insider attack higher than a given threshold.

Although game-theoretic access control and temporal-logic-based policy specification have been investigated in prior research, the novelty of RAFIA does not lie in introducing these techniques independently. Rather, the contribution of RAFIA is the integration of quantitative risk estimation, trace-dependent utility computation, multi-stage game-theoretic reasoning, and Risk-LTL policy enforcement within a unified framework for insider-threat mitigation. Unlike conventional risk-based access control approaches that evaluate requests independently, RAFIA incorporates the history of accepted and denied actions into risk assessment and access-control decisions. Furthermore, Risk-LTL serves as a risk-aware policy layer that constrains authorization decisions according to dynamically computed risk values over execution traces.

The proposed contributions directly address the research questions as follows. Contribution C2 addresses RQ1 by introducing Risk-LTL, a temporal logic for specifying insider-attack-related security policies. Contribution C1 addresses RQ2 through a formal risk-assessment approach that combines attack probability, impact evaluation, and trace-based risk assessment. Contribution C3 addresses RQ3 through RAFIA, which integrates risk estimation, trace-based analysis, and game-theoretic decision making to support the detection and prevention of insider attacks.

The rest of this paper is organized as follows: Section 2 provides an overview of the proposed approach. Section 3 introduces relevant preliminaries. In Section 4, we present the modeling of the access control system using game theory, followed by the enhanced version of the proposed approach in Section 5. Section 6 explains the evaluation of risk related to a trace, while Section 7 models access control as a multi-stage game. The syntax and the semantics of the Risk-LTL are defined in Section 8. Relevant related works and comparison of the proposed approach with existing methods are presented in Section 9. Section 10 presents the prototype implementation and its experimental evaluation. Section 11 discusses RAFIA's integration of dynamic risk assessment with adaptive access control to enhance security and usability. Section 12 concludes the paper and outlines directions for future work.

2. Proposed Risk Estimation Model

For any organization, it is essential to have an access control system that ensures adequate security protection for the resources according to a given security policy. The impact of malicious access can be so great that it could even lead to the bankruptcy of the organization. To avoid such situations, an appropriate risk model to evaluate the risk related to any access control query should be set up. To estimate the risk, a number of factors should be taken into consideration, such as the trust levels of users, the sensitivities of data, the history of accesses, and the context. These factors influence the risk assessment and hence the decision-making process. Clearly, risk estimation represents the essence and core of the risk-based access control model.

As defined in many works, such as [15,16], the risk is the multiplication of probability by impact, as shown by the following formula:

$$R = P(A) \times I(A), \quad (1)$$

where $P(A)$ is the probability of an attack A and $I(A)$ is its impact. When evaluating the impact of an attack, we take into consideration the three main security properties: Confidentiality, Integrity and Availability which may sometimes overlap, as shown in Fig. 2.

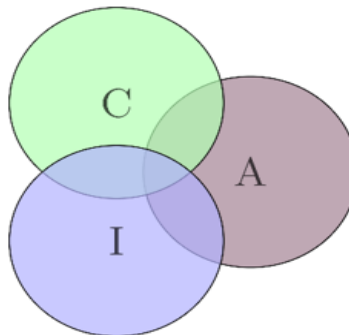


Fig. 2. The CIA security properties.

Therefore, having an appropriate method to estimate both the probability of an attack and its potential impact is essential in order to avoid blocking legitimate access while also preventing the authorization of malicious activity. The

proposed risk estimation model provides a practical approach for estimating the risk associated with an access query, along with an expressive policy language for specifying security policies. Fig. 3 summarizes the approach, and its main steps are as follows:

- The system is initialized by the security manager during step (1) by providing its security policy Φ , specified in a linear temporal logic called Risk-LTL and defined in Section 8. The manager also provides the risk function R , which is defined based on the impact of each action performed on the system, as well as the probability that a given action is executed by a malicious user. The impact on different security properties (Confidentiality, Integrity, and Availability) is evaluated based on the value of the asset (object) and the type of action (read, write, etc.). However, the most challenging part is evaluating the probability that an action is malicious and identifying the users who could profit from it.

In RAFIA, this probability is represented by the maliciousness function $\rho(s)$, introduced formally in Definition 2 and subsequently incorporated into the risk computation and game-theoretic decision process.

- Once the access control is initialized, it can receive requests from users at step (2). Each action contains the name a (for example, read, write, etc.), the subject s and the object o .
- At step (3), the access control system starts analyzing the action. It takes Φ and R from the storage (step (4)), the context c from the context provider module (step (5)). After that, it evaluates whether the action $a(s, o, c) \models_R \Phi$. The definition of $\models_R$ is given by the semantics of the logic in Section 8. The formula can involve risk such as $Risk_{<\lambda}(\Phi)$ which intuitively means that the action $a(s, o, c)$ has to respect the security policy Φ and the risk associated with it, $R(a(s, o, c))$, should be lower than a threshold λ .
- If the evaluation of a formula is positive, the action is communicated to the system (step (7)) to be executed; otherwise, it is blocked.

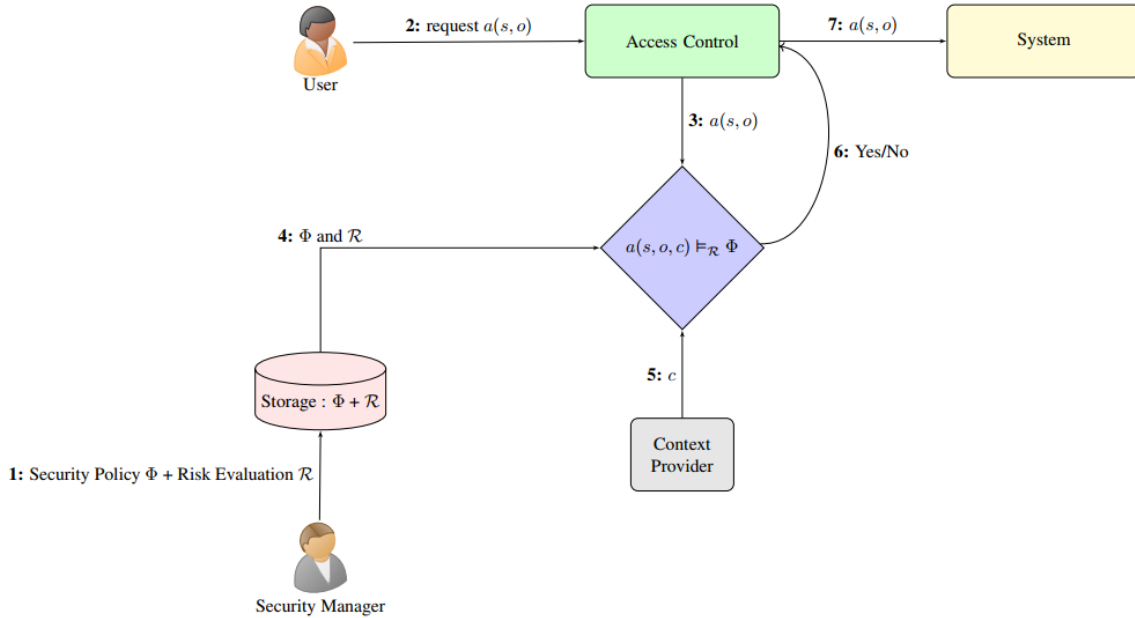


Fig. 3. Risk Estimation Model.

3. Preliminaries

To simplify the presentation of the proposed approach, we introduce the notations used throughout the paper. We assume the existence of a set of subjects $\mathcal{S}$ representing users who wish to access a set of objects $\mathcal{O}$ by performing actions from a set $\mathcal{A}$ within a set of contexts $\mathcal{C}$ representing real-time contextual information, such as time, IP address, and location. These sets together define the access-control domain.

Definition 1 (Access Control Domain (ACD)). *An Access Control Domain (ACD), denoted by*

$$\Sigma = \langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{C} \rangle,$$

is a quadruple where

- $\mathcal{S}$ is a finite set of subjects;
- $\mathcal{O}$ is a finite set of objects;
- $\mathcal{A}$ is a finite set of actions; and
- $\mathcal{C}$ is a finite set of contexts.

A risk model within a given access control domain assesses the potential impact of each action and the probability that it is performed by a malicious user.

Definition 2 (Risk Model (RM)). *A Risk Model (RM), denoted by*

$$RM = \langle \Sigma, \iota, \rho \rangle,$$

is defined as a triple, where:

- $\Sigma = \langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{C} \rangle$ is an Access Control Domain (ACD);
- $\iota : \mathcal{A} \times \mathcal{S} \times \mathcal{O} \times \mathcal{C} \rightarrow \mathbb{R}$ is an impact function; and
- $\rho : \mathcal{S} \rightarrow [0, 1]$ is the maliciousness function that assigns to each subject $s \in \mathcal{S}$ the probability that the subject behaves maliciously.

4. Game-Theoretic Access Control Model

To detect insider attacks—where malicious users strategically adapt their behavior to evade detection—we model access control as a strategic game between the organization (defender) and users (potential attackers). Within RAFIA, game-theoretic reasoning enables the following enhancements:

1. **Embedding $R(a)$ in payoff matrix:** Formulating the risk function $R(a)$ as an adversarial payoff function, where malicious actors strategically optimize their actions to maximize illicit benefits while constraining their risk exposure below detection thresholds.
2. **Dynamic Defense Thresholds:** Reformulating the threshold λ as an equilibrium-driven adaptive parameter that evolves in response to optimal attacker strategies.
3. **Context-Aware Protection:** Formalizing context (c) as part of the action space, where game states evolve according to observable contextual attributes such as {time, location, behavior}.

This transforms access control from fixed rules to intelligent adaptive security systems that learn attacker behavior and automatically strengthen defenses. Having established this game-theoretic framework, we can now examine how it advances traditional access control principles. Access control mechanisms serve as a fundamental security technique to mitigate unauthorized access attempts by malicious users. However, the issue of security and preventing unauthorized access remains a challenge. Access control mechanisms often rely on predetermined rules or statistical models to make decisions. In high-stakes or uncertain environments, a probabilistic decision-theoretic approach may be more appropriate. Accordingly, access control can be modeled as a game between users and companies, as game theory provides rigorous mathematical and conceptual tools to model strategic interactions, predict outcomes (equilibria), analyze vulnerabilities, and design resilient systems. One of its key strengths is the ability to evaluate multiple scenarios before selecting an optimal strategy. Accordingly, this work models the decision to accept or reject a user's action as a game involving uncertain user behavior and known utility outcomes.

Before proceeding with implementation, we must ground this model in its theoretical foundations [17]. Game theory offers a formal analytical framework for examining and simulating strategic choices within dynamic, unpredictable settings [17]. A key benefit is its capacity to evaluate multiple hypothetical situations prior to determining an optimal approach. This aligns well with access control mechanisms, where user-organization interactions can be interpreted as a competitive decision-making process. Within this framework, participants—usually individuals and the system administrators—select varying tactics that shape the final results. By analyzing these choices and their associated consequences, the approach reveals critical vulnerabilities and supports the development of more resilient access management protocols.

Fundamentally, any game-theoretic model consists of basic elements, which are: the players (the fundamental components of the game—decision-makers such as users, machines, or information systems); strategies (the plans of action chosen by users to play the game); actions (the choices made by each player at each stage of the game); payoff (the outcome for each player based on the combination of chosen strategies); and equilibrium (the solution or stable outcome of the game).

4.1. One-Shot-Game: Model Description

This model assumes rational behavior from both the user and the company, framing the access control mechanism as a one-shot strategic game, as illustrated in Fig. 4. The game involves two participants, each with a defined set of strategies and associated payoff functions.

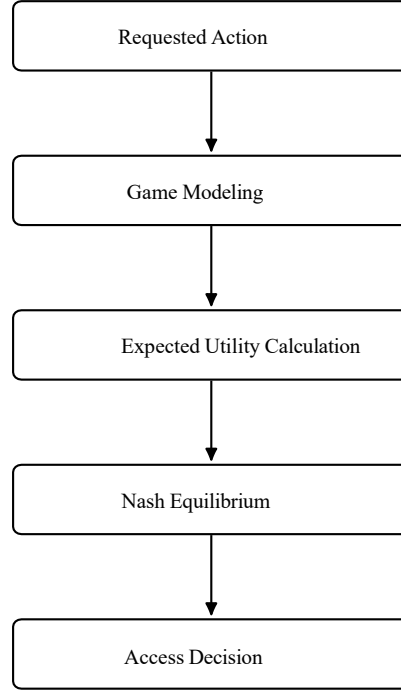


Fig. 4. One-Shot Game-Based Access Control.

1. **Access Request Initiation:** The user initiates an access request by specifying their access request.
2. **Strategic Game Formulation:** The system formulates the interaction as a strategic one-shot game to evaluate decisions in a single, isolated context between two players:
 - **Player 1:** The user (U), who seeks to access resources, can choose between two strategies: performing an *Honest Access* (H) or performing a *Malicious Access* (M).
 - **Player 2:** The company (C), which owns the sensitive resources, has two strategic options: *Accept* (A) or *Deny* (D).
3. **Expected Utility:** Let q be the probability that the user acts honestly, and $1 - q$ the probability that the user acts maliciously. Let p denote the probability that the company accepts the action and $1 - p$ the probability that the company rejects the action.
 - **Company Utilities**
 - $U_c^{(A,H)}$: Utility for accepting an honest action.
 - $U_c^{(A,M)}$: Utility for accepting a malicious action.
 - $U_c^{(D,H)}$: Utility (negative) for denying an honest action.
 - $U_c^{(D,M)}$: Utility for denying a malicious action.

Accordingly, the expected utility of the company when accepting a request is given by

$$\mathbb{E}[U_c | A] = qU_c^{(A,H)} + (1 - q)U_c^{(A,M)} \quad (2)$$

Similarly, when the company denies the request:

$$\mathbb{E}[U_c | D] = qU_c^{(D,H)} + (1 - q)U_c^{(D,M)} \quad (3)$$

- **User Utilities**

- $U_u^{(A,H)}$: Utility for the honest user if accepted.
- $U_u^{(D,H)}$: Loss for the honest user if denied.
- $U_u^{(M,A)}$: Gain for the malicious user if accepted.
- $U_u^{(M,D)}$: Loss for the malicious user if denied.

Consequently, the expected utility of the user is given by

$$\begin{aligned} \mathbb{E}[U_u] = & pqU_u^{(A,H)} + (1-p)qU_u^{(D,H)} + \\ & p(1-q)U_u^{(M,A)} + (1-p)(1-q)U_u^{(M,D)} \end{aligned} \quad (4)$$

The payoff functions for each party, based on the combination of chosen strategies, are presented in Table 1.

Table 1. Payoff matrix of the one-shot game

	Accept(p)	Deny($1-p$)
Honest (q)	$(U_u^{(A,H)}, U_c^{(A,H)})$	$(U_u^{(D,H)}, U_c^{(D,H)})$
Malicious ($1-q$)	$(U_u^{(M,A)}, U_c^{(M,A)})$	$(U_u^{(M,D)}, U_c^{(M,D)})$

- **Row Player:** User (chooses between *Honest* or *Malicious* behavior)
- **Column Player:** Company (chooses *Accept* or *Deny*)
- **Payoffs:** (User's utility, Company's utility). In each cell, the first utility value is for the User, the second is for the Company
- **(Honest, Accept):** User: $U_u^{(A,H)}$, Company: $U_c^{(A,H)}$
- **(Honest, Deny):** User: $U_u^{(D,H)}$, Company: $U_c^{(D,H)}$
- **(Malicious, Accept):** User: $U_u^{(M,A)}$, Company: $U_c^{(M,A)}$
- **(Malicious, Deny):** User: $U_u^{(M,D)}$, Company: $U_c^{(M,D)}$

4. Nash Equilibrium Computation:

We now analyze the interaction as a simultaneous-move game. The user chooses q (honesty level), and the company chooses p (acceptance probability). Each player's goal is to maximize their expected utility. Nash equilibrium theory is a central concept in game theory, which means that each player is aware of the opponent's strategies and does not intend to change his strategy because it does not maximize his payoff.

To analyze the access control scenario, we model the interaction between the user and the company as a simultaneous action game, where the user chooses a strategy $q \in [0, 1]$, representing the probability of acting honestly and the company chooses a strategy $p \in [0, 1]$, indicating the probability of granting access, and both players act rationally as their goal is to maximize their expected utility. In this context, a Nash Equilibrium (NE) occurs when neither party can increase their expected payoff by changing their strategy unilaterally. At equilibrium, each strategy is a best response to the other.

The best response functions for both players are defined as follows:

- The company's best response $p^*(q)$ is:

$$p^*(q) = \begin{cases} 1 & \text{if } \mathbb{E}[U_c | A] > \mathbb{E}[U_c | D] \\ 0 & \text{if } \mathbb{E}[U_c | A] < \mathbb{E}[U_c | D] \\ \text{any } p \in [0, 1] & \text{if } \mathbb{E}[U_c | A] = \mathbb{E}[U_c | D] \end{cases} \quad (5)$$

- The user's best response $q^*(p)$ is:

$$q^*(p) = \begin{cases} 1 & \text{if } p(U_u^{(A,H)} - U_u^{(D,H)}) > \\ & (1-p)(U_u^{(M,A)} - U_u^{(M,D)}) \\ 0 & \text{if the inequality is reversed} \\ \text{any } q \in [0, 1] & \text{if both sides are equal} \end{cases} \quad (6)$$

Solving these coupled best response functions yields the Nash Equilibrium strategy profile(p^*, q^*), which represents a stable and rational decision outcome for both parties in the access control system.

5. Equilibrium-Driven Decision:

The company enforces its equilibrium strategy as the access decision:

$$\text{Decision} = \begin{cases} \text{Grant, if the Nash equilibrium strategy is Accept,} \\ \text{Deny, if the Nash equilibrium strategy is Deny.} \end{cases} \quad (7)$$

The company accepts the action if

$$\mathbb{E}[U_c | A] > \mathbb{E}[U_c | D], \quad (8)$$

where $\mathbb{E}[U_c | A]$ and $\mathbb{E}[U_c | D]$ denote the expected utilities of accepting and denying the access request, respectively.

Substituting the expected utility expressions yields

$$q(U_c^{(A,H)} - U_c^{(D,H)}) + (1 - q)(U_c^{(A,M)} - U_c^{(D,M)}) > 0, \quad (9)$$

where q represents the probability that the user is honest.

Solving for q gives the acceptance threshold:

$$q > \frac{U_c^{(D,M)} - U_c^{(A,M)}}{(U_c^{(A,H)} - U_c^{(D,H)}) + (U_c^{(D,M)} - U_c^{(A,M)})} \quad (10)$$

4.1.1. Numerical Examples

To better understand the practical implications of modeling the access control decision process as a one-shot game between a **user** and a **company**, we now present numerical examples. These examples illustrate how the Nash Equilibrium can be derived by analyzing the strategic interaction between the two parties.

Example 1. Consider the payoff matrix shown in Table 2:

Table 2. Payoff matrix for Example 1.

	Accept (p)	Deny ($1 - p$)
Honest (q)	(5, 5)	(1, 3)
Malicious ($1 - q$)	(8, -10)	(0, 4)

Step 1: Expected Utility for the Company

- If the company **accepts** the request,

$$\begin{aligned} \mathbb{E}[U_c | A] &= q \cdot 5 + (1 - q) \cdot (-10) \\ &= 15q - 10. \end{aligned}$$

- If the company **denies** the request,

$$\mathbb{E}[U_c | D] = q \cdot 3 + (1 - q) \cdot 4 = 4 - q.$$

Step 2: Expected Utility for the User

$$\begin{aligned} \mathbb{E}[U_u] &= pq \cdot 5 + (1 - p)q \cdot 1 + \\ &+ p(1 - q) \cdot 8 + (1 - p)(1 - q) \cdot 0 \\ &= 5pq + q(1 - p) + 8p(1 - q) \\ &= q - 4pq + 8p. \end{aligned}$$

Summary of Expected Utilities

$$\mathbb{E}[U_c | A] = 15q - 10,$$

$$\mathbb{E}[U_c | D] = 4 - q,$$

$$\mathbb{E}[U_u] = q - 4pq + 8p.$$

These formulas help each player decide their best strategy based on the estimated behavior of the other. The company will compare $\mathbb{E}[U_c | A]$ and $\mathbb{E}[U_c | D]$ to choose whether to accept or deny, while the user considers $\mathbb{E}[U_u]$ to decide whether to act honestly or maliciously.

According to the proposed model, the company accepts the request if

$$\mathbb{E}[U_c | A] > \mathbb{E}[U_c | D]$$

Expanding and applying the specific utility values:

$$\begin{aligned} & q(U_c^{(A,H)} - U_c^{(D,H)}) + \\ & (1 - q)(U_c^{(A,M)} - U_c^{(D,M)}) > 0 \\ & q(5 - 3) + (1 - q)(-10 - 4) > 0 \\ & 2q + (1 - q)(-14) > 0 \\ & 2q - 14 + 14q > 0 \Rightarrow 16q > 14 \\ & q > \frac{14}{16} = 0.875. \end{aligned}$$

The company should accept access requests only if it believes the user is honest with probability greater than 0.875.

The user chooses to behave honestly if:

$$p(U_u^{(A,H)} - U_u^{(D,H)}) > (1 - p)(U_u^{(M,A)} - U_u^{(M,D)})$$

Substituting the numerical payoffs:

$$\begin{aligned} & p(5 - 1) > (1 - p)(8 - 0) \\ & 4p > 8(1 - p) \\ & 4p > 8 - 8p \Rightarrow 12p > 8 \\ & p > \frac{8}{12} = \frac{2}{3} \approx 0.667 \end{aligned}$$

The user will act honestly only if the company is likely to accept the request with probability greater than $\frac{2}{3}$. Given the thresholds derived above:

- The company accepts only if it believes the user is honest with probability $q > 0.875$.
- The user behaves honestly only if the company is expected to accept with probability $p > \frac{2}{3}$

These coupled conditions indicate that a Nash Equilibrium occurs when both thresholds are simultaneously satisfied:

$$p^* > \frac{2}{3}, \quad q^* > 0.875$$

This means that in equilibrium:

- The user will be incentivized to act honestly because the company is likely to accept requests.
- The company will accept requests because the user is likely honest.

The equilibrium is only sustained when mutual trust is high. If either player deviates (e.g., the user becomes more malicious, or the company becomes overly cautious), the equilibrium collapses, leading to denial or attack outcomes.

This numerical analysis illustrates the strategic dependencies between user honesty and company acceptance within an access control system modeled as a one-shot game. The derived thresholds— $p > \frac{2}{3}$ for the user and $q > 0.875$ for the company—define the conditions under which mutual cooperation (i.e., access granted and honest behavior) becomes a stable outcome in the form of a Nash Equilibrium. This equilibrium region emphasizes the importance of trust calibration: the system must assess user credibility accurately and assign access rights accordingly.

Example. 2 Alternative Payoff Structure

Step 1: Expected Utility for the Company

The payoff matrix used in Example 2 is presented in Table 3

Table 3. Payoff matrix for Example 2.

	Accept (p)	Deny ($1 - p$)
Honest (q)	(3, 5)	(-5, -2)
Malicious ($1 - q$)	(8, -10)	(-12, 0)

Step 2: Expected Utility for the Company

- If the company **accepts** the request:

$$\mathbb{E}[U_c | A] = q \cdot 5 + (1 - q) \cdot (-10) = 15q - 10.$$

- If the company **denies** the request:

$$\mathbb{E}[U_c | D] = q \cdot (-2) + (1 - q) \cdot 0 = -2q$$

Step 3: Expected Utility for the User

$$\begin{aligned} \mathbb{E}[U_u] &= pq \cdot 3 + (1 - p)q \cdot (-5) + \\ & p(1 - q) \cdot 8 + (1 - p)(1 - q) \cdot (-12) \\ &= 3pq - 5q(1 - p) + 8p(1 - q) - 12(1 - p)(1 - q) \end{aligned}$$

Step 3: Threshold Calculation

Company's Decision Threshold

The company accepts if:

$$\mathbb{E}[U_c | A] > \mathbb{E}[U_c | D]$$

$$15q - 10 > -2q$$

$$17q > 10$$

$$q > \frac{10}{17} \approx 0.588$$

User's Decision at Equilibrium: $p = 0.6$

Honest behavior is preferable if:

$$0.6(3 - (-5)) > 0.4(8 - (-12))$$

$$0.6 \cdot 8 = 4.8, 0.4 \cdot 20 = 8$$

$$4.8 < 8 \Rightarrow \text{False}$$

Therefore, the user prefers the malicious strategy when $p = 0.6$.

4.2. Limitations of One-Shot-Game

To connect theory with real-world understanding, the earlier numerical examples illustrate how a user and a company interact strategically within a one-shot game-based access control model. These examples not only demonstrate the derivation and interpretation of Nash equilibrium thresholds based on specific payoff structures but also reveal the conditions under which cooperation—i.e., honest user behavior and access approval—can be sustained. The numerical results help show how a one-shot game model can support decision-making in situations where uncertainty is high and rapid decisions are needed.

However, these examples also bring out some of the model's key weaknesses. One-shot game models do not consider how users' behavior can change gradually. They treat every decision as if it is happening without any context or history. But in most real systems, behavioral history plays a critical role. If a user's behavior starts to change, even slowly, a model that doesn't track that change won't respond until it's too late, which limits their effectiveness in environments where user behavior is rarely static [18]. The following section explores the shortcomings of such one-shot game models in more detail and recommends the use of multi-stage game-based access control models that can respond to behavioral changes and take history into account for granting access decisions. The key limitations are:

4.2.1. Memoryless Evaluation: Lack of Historical Context and Temporal Dynamics

One-shot access control models suffer from *atomic decision-making* – each access request is evaluated in isolation, without considering past interactions or behavioral trends. This manifests in three critical flaws:

a) No Historical Context

The system cannot correlate actions over time. For example, in Canada's Desjardins Group breach [19], an employee extracted sensitive data gradually over two years. Because each access was evaluated independently, the system failed to detect the escalating pattern.

b) Static Strategies

Fixed trust thresholds (e.g., "grant access if score > 0.6 ") cannot adapt to evolving threats. Table 4 shows how a user's trust decays slowly ($0.80 \rightarrow 0.59$), but access is granted until the final suspicious download.

c) Ignored Temporal Patterns

Sequential attacks (e.g., the Carnegie Mellon insider threat [20]) exploit this. A user who starts with minor policy violations (e.g., occasional unauthorized file access) and escalates gradually remains undetected because one-shot models lack time-series analysis.

Table 4. Limitations of static thresholds in one-shot models.

Round	Behavior	Trust Score	Decision
1	Normal access	0.80	Grant
2	Slight anomaly	0.75	Grant
3	Sensitive data access	0.68	Grant
4	Late-night access	0.60	Grant
5	Suspicious download	0.59	Deny

4.2.2. Exploitability—Fixed Acceptance Thresholds Enable Repeated Malicious Exploitation

A fixed acceptance threshold is one of the main weaknesses of one-shot game-based access control models. The access system grants or denies requests based on the user's perceived level of trust, so a malicious user exploits the fixed threshold to adapt their behavior to stay just above the threshold and repeatedly exploit the system without being detected. A relevant real-world case is the 2019 Capital One data breach [16]. The attacker was able to exploit a misconfigured firewall to gain credentials and access sensitive data stored in the cloud. Furthermore, the access control mechanisms in place did not adapt or respond to the unusual activity that followed. What enabled the attacker to move within the system

undetected was that the rules governing access were static and did not take into account any change in user behavior over time. Although the breach was attributed to a technical flaw, the lack of a dynamic response allowed it to escalate. This is similar to what happens in one-shot game-based access systems, where the user is initially trusted, this decision is not reviewed and the user begins to change their behavior until he exploits this weakness and behaves dishonestly. Therefore, it is important to design access control models that can adapt to changes occurring over time, rather than relying on fixed assumptions about trust, as is the case with one-shot game-based access control models. Access control models based on multi-stage games provide better solutions to mitigate these risks by tightening controls when suspicious behavior patterns appear.

To demonstrate the limitation of fixed acceptance thresholds in one-shot access control games, consider the following scenario. An insider user attempts to repeatedly access sensitive information by maintaining their observed behavior just above a system's fixed threshold value. The system, in this case, applies a fixed threshold value (e.g., 0.7) to determine whether access should be granted.

In this setup, the user's trust level is kept constant at 0.71, which is slightly above the threshold. Since the policy does not adjust according to the user's past behavior, a malicious user can exploit this loophole and repeatedly bypass the trust threshold without being detected. To address this vulnerability, a dynamic threshold model is proposed where the system incrementally raises the access threshold based on previous access attempts. Thus, even if the attacker's trust level remains unchanged, the access decisions are increasingly stringent and eventually the access request is denied once it exceeds the trust level of the updated threshold. Fig. 5 presents the difference between the dynamic threshold and fixed threshold policies. The blue line represents the fixed threshold, which does not adapt to repeated access attempts. An attacker who maintains a constant trust level just above this threshold continues to receive access without being detected. On the other hand, the dynamic threshold in red increases over time. Although the attacker's behavior remains unchanged, the system becomes progressively stricter. Once the threshold exceeds the attacker's trust level, the access request is denied. This illustrates how dynamic thresholding reduces the risk of repeated malicious access and motivates the adoption of multi-stage rather than one-shot access control models.

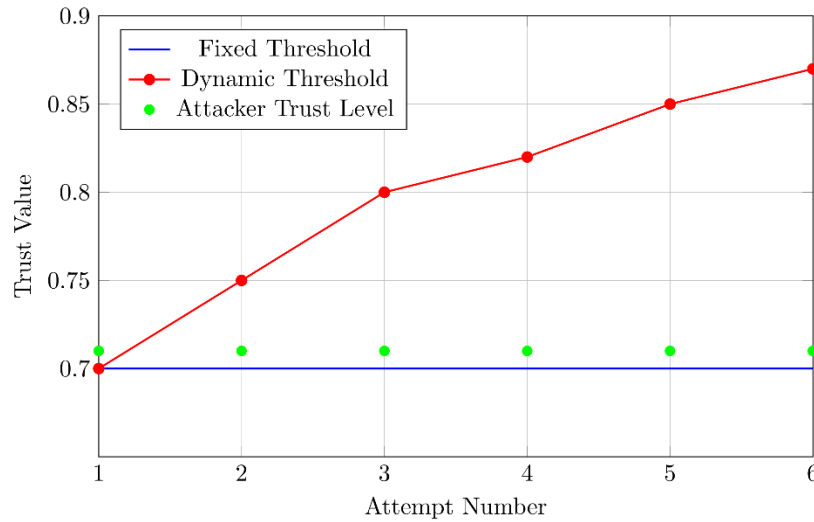


Fig. 5. Comparison between fixed and dynamic trust thresholds across repeated access attempts.

4.2.3. Multi-Stage Game Solutions

Multi-stage game models overcome these limitations through the following mechanisms:

History-Aware Policies: Track user actions over time (e.g., Bayesian trust updates)

Dynamic Thresholds: Adjust access rules based on recent behavior

Sequential Analysis: Compute cumulative utility:

$$U_{cumulative} = \sum_{t=1}^T \mathcal{R}(t) - \mathcal{C}(t), \quad (11)$$

where $\mathcal{R}(t)$ is the reward and $\mathcal{C}(t)$ is the cost at time t .

From this perspective, we conclude that using a multi-stage game to model access control systems is a stronger and more effective approach than using a one-shot game for several reasons. First, these models adapt to the user's behavior

changes over time, allowing decisions to be made based on the user's history and evolving behaviors rather than on a single instantaneous assessment without considering behavioral history. Second, multi-stage models reduce the risk of repeated attacks by malicious users through gradually raising security thresholds or changing strategies. Third, they provide a continuous mechanism to evaluate the user's trust level, which evolves according to their behavior, thus reflecting a more accurate assessment of potential risks. Fourth, the system leverages the ongoing collection and dynamic evaluation of data to make smarter and more efficient decisions about access control. It also supports integrating learning and adaptation methods, which enable the system to progressively enhance its security measures as user behaviors evolve. Lastly, these models accurately reflect the strategic back-and-forth between the user and the system, with both parties adjusting their actions step-by-step, making them especially effective for addressing complex and ever-changing cyber threats.

5. RAFIA: Overview

To support multi-stage game-theoretic modeling, the proposed risk estimation model presented in Fig. 3. is extended by incorporating the history of previous user actions into the access-control decision process. The resulting framework, referred to as RAFIA, is a game-theoretic risk-based framework for insider-threat mitigation. Fig. 6. provides an overview of RAFIA, and its main processing steps are described as follows.

- The system is initialized by the security manager during step (1) by providing its security policy Φ , specified in a linear temporal logic called Risk-LTL and defined in Section 8. The security manager also provides the risk function R , which is defined based on the impact of each action performed on the system, as well as the probability that a given action is executed by a malicious user. The impact on different security properties (Confidentiality, Integrity, and Availability) is evaluated based on the value of the asset (object) and the type of action (read, write, etc.). However, the most challenging part is evaluating the probability that an action is malicious and identifying the users who could profit from it.
- Once the access control is initialized, it can receive requests from users at step (2). Each action contains the name a (for example, read, write, etc.), the subject s and the object o .

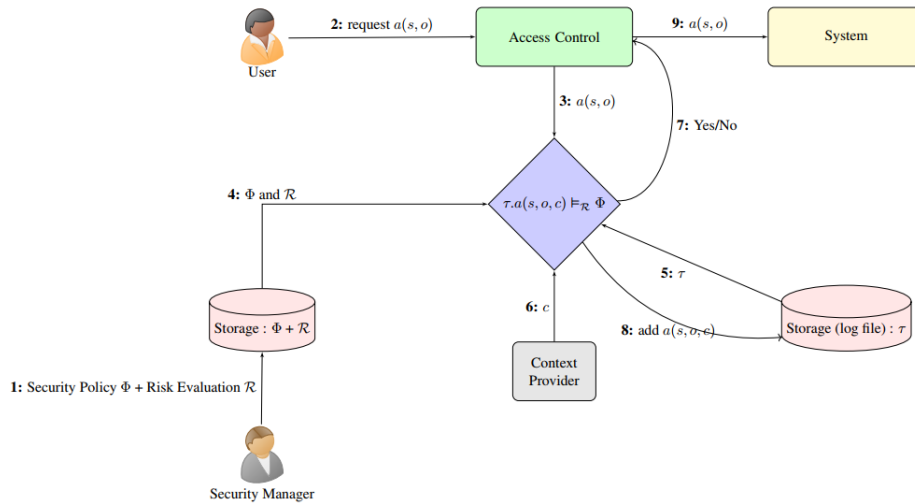


Fig. 6. Overview of the Proposed RAFIA Framework.

- At step (3), the access control system starts analyzing the action. It takes Φ and R from the storage (step (4)), the context c from the context provider module (step (6)) and the trace τ (containing the previously executed actions) from the storage (step (5)). After that, it evaluates if $\tau.a(s, o, c) \models_R \Phi$. The definition of $\models_R$ is given by the semantics of the logic in Section 8. The formula can involve risk such as $Risk_{<\lambda}(\Phi)$ which intuitively means that the trace $\tau.a(s, o, c)$ has to respect the security policy Φ and the risk associated with it, $R(\tau.a(s, o, c))$, should be lower than a threshold λ .
- If the evaluation of a formula is positive, the action is communicated to the system (step (9)) to be executed; otherwise, it is blocked. Moreover, the trace is updated by adding accepted actions (step (8)).

Deciding if a query is acceptable or not should take into consideration not only the query itself but also all the previous activities (trace) performed on the system. For example, a security policy may limit the number of records of certain types that a specific user can access. In this case, the trace (history) is needed to evaluate a request. Hereafter, we introduce some notation related to traces.

Definition 3. *[Trace]*

Let $\Sigma = \langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{C} \rangle$ be an ACD.

- A trace over Σ (or simply a trace when the ACD is clear from the context) is defined using the following BNF grammar:

$$\tau ::= \epsilon \mid a(s, o, c) \mid a(s, o, c). \tau$$

where $(s, o, a, c) \in \mathcal{S} \times \mathcal{O} \times \mathcal{A} \times \mathcal{C}$, ϵ is an empty trace and $a(s, o, c). \tau$ is the concatenation of the action $a(s, o, c)$ with the trace τ .

- Set of possible traces ($\mathcal{T}_\Sigma$): We denote by $\mathcal{T}_\Sigma$ the set of possible traces built from Σ .
- Suffix of a trace: If $\tau = a_0.a_1.a_2 \dots$ is a trace, then $\tau(i) = a_i$ and $\tau^j = a_j.a_{j+1} \dots$
- Size of a trace ($|\tau|$): The size of a trace gives the number of actions in it. More formally, the size of a trace τ , will be denoted by $|\tau|$, is defined as follows:

$$|\epsilon| = 0$$

$$|a.\tau| = 1 + |\tau|$$

- Projection of a trace on a set of subjects ($\tau[S]$): If τ is a trace and S is a set of subjects, we define the projection of τ on S (the set of actions in which a subject in S is involved in τ), denoted by $\tau[S]$, inductively as follows:

$$\begin{aligned} \epsilon[S] &= \{\} \\ a(s, o, c)[S] &= \{\} \text{ if } s \notin S \\ a(s, o, c)[S] &= \{a(s, o)\} \text{ if } s \in S \\ (\tau.\tau')[S] &= \tau[S] \cup \tau'[S] \end{aligned}$$

- Projection of a trace on a set actions ($\tau_{\downarrow A}$): If τ is a trace and A is a set of actions, we define the projection of τ on A , denoted by $\tau_{\downarrow A}$, inductively as follows:

$$\begin{aligned} \epsilon_{\downarrow A} &= \epsilon \\ (a.\tau)_{\downarrow A} &= a.(\tau)_{\downarrow A} \text{ if } a \in A \\ (a.\tau)_{\downarrow A} &= (\tau)_{\downarrow A} \text{ if } a \notin A \end{aligned}$$

if $A = \{a\}$, $\tau_{\downarrow \{a\}}$ will be simply abbreviated by $\tau_{\downarrow a}$ and $\tau_{\downarrow -a}$ will be an abbreviation of $\tau_{\downarrow A - \{a\}}$.

- Renaming a subject ($\tau[s \mapsto s']$): We can rename any subject, object or action in a trace. To rename a subject s by s' in τ , denoted by $\tau[s \mapsto s']$, we proceed according to the following rules:

$$\begin{aligned} \epsilon[s \mapsto s'] &= \epsilon \\ a(s, o, c)[s \mapsto s'] &= a(s', o, c) \\ a(s'', o, c)[s \mapsto s'] &= \epsilon \text{ if } s'' \neq s \\ (a(s'', o, c).\tau)[s \mapsto s'] &= \\ & a(s'', o, c)[s \mapsto s'].\tau[s \mapsto s'] \end{aligned}$$

A similar definition could be given to rename objects or actions.

- Renaming a set of subjects ($\tau[S \mapsto s']$): The previous definition could be extended to a set of subjects S , denoted by $\tau[S \mapsto s']$, as follows:

$$\begin{aligned} \tau[\emptyset \mapsto s'] &= \tau \\ \tau[\{s\} \mapsto s'] &= \tau[s \mapsto s'] \\ \tau[\{s\} \cup S \mapsto s'] &= (\tau[s \mapsto s'])[S \mapsto s'] \end{aligned}$$

6. Risk Related to a Trace

When a sequence of actions is executed inside a system, there is a risk that some of them are executed by malicious subjects. This risk depends on the probabilities that actions are malicious and the impact of these actions. The risk related to any action should not exceed a threshold.

In this section, we explain the impact of a trace, the probability of an attack and finally we define the risk associated with a given trace.

6.1. Impact

When we compute the impact related to the actions in which a given subject is involved, we need to take several facts into account.

When an insider attack occurs in a company, it can affect both the company and the individuals involved. We need to understand how to quantify these impacts in order to effectively evaluate the risk of insider attacks.

6.1.1. Company Side

When an action is executed maliciously, it may have an impact on one or many security properties: Availability (A), Confidentiality (C) and Integrity (I) and we should add the impact related to all of them.

- **Availability:** Actions such as write or delete can make some data unavailable. Also, other actions like stop (a hardware), reboot (a hardware), close (a port), or kill (a process) can make some services or data un-available. Therefore the impact of an action $a(s, o, c)$ depends on the object and can depend on the context. For example, stopping the service during the night or on the weekend would have less impact than during a business day. Furthermore, if the same action is perpetrated more than one time, the impact could increase.
- **Integrity:** Actions such as write or modify can affect the integrity of data. Most of the time the impact is independent of context and if the same action is repeated, would not increase the impact.
- **Confidentiality:** Actions such as read or send can affect the confidentiality of data. Most of the time the impact is independent of context and if the same action is repeated, the impact will not increase.

The impact function should answer at least the following questions:

- **What is the impact (payoff) of a single action?** From the company's point of view, what is the impact of each individual action on every security property (Confidentiality, Integrity, and Availability)? The impact function should provide both the positive and negative consequences of executing any malicious action on any object. Whether an action is accepted or refused, there are both positive and negative impacts. Of course, while refusing all actions would eliminate insider attacks, it would not be a viable strategy. When an action is accepted, the potential benefit to the company is U_{AH}^c (true positive impact) and the possible loss is U_{AM}^c (false positive impact) and when an action is refused, the positive impact is U_{DM}^c (false negative impact) and the negative impact is U_{DH}^c (true negative impact).

For every action and every security property, the company needs to provide an impact table as shown by Table 5.

Table 5. Payoff matrix: action versus impact.

	Accept (A)	Deny (D)
Honest (H)	U_{AH}^c	U_{DH}^c
Malicious (M)	U_{AM}^c	U_{DM}^c

The impact should take into consideration the different security properties. For an action a and a security property P , we use $U_{AH}^c(a, P)$, $U_{DH}^c(a, P)$, $U_{AM}^c(a, P)$ and $U_{DM}^c(a, P)$ to denote, respectively, the impact of accepting an honest action, denying an honest action, accepting a malicious action and denying a malicious action. If the security property is not specified, it implicitly means that we are dealing with the cumulative impact as formalized by the following definition.

Definition 4. ($U_{AH}^c(a)$, $U_{AM}^c(a)$, $U_{DH}^c(a)$ and $U_{DM}^c(a)$). Let a be an action. We define $U_{AH}^c(a)$, $U_{AM}^c(a)$, $U_{DH}^c(a)$ and $U_{DM}^c(a)$ as follows:

$$U_{AH}^c(a) = \sum_{P \in \{C, I, A\}} U_{AH}^c(a, P)$$

$$U_{AM}^c(a) = \sum_{P \in \{C, I, A\}} U_{AM}^c(a, P)$$

$$U_{DH}^c(a) = \sum_{P \in \{C, I, A\}} U_{DH}^c(a, P)$$

$$U_{DM}^c(a) = \sum_{P \in \{C, I, A\}} U_{DM}^c(a, P)$$

Notice that usually $U_{DM}^c(a) = 0$.

- **What is the impact of a trace?** Knowing the impact of every action would not be enough to evaluate the impact of a trace (sequence of actions). Here are some reasons:
 - Action repetition: In some situations, doing the same action many times does not necessarily multiply the impact (e.g. read a private information) but in other situation the impact can be increased (e.g. the impact of the denial of service of one hour could be different for two or three hours.)
 - Action ordering: The penalties and the benefits of running $a.b$ could be different for those of $b.a$. For instance if a is reading a secret data and b corresponds to sending some data on the Internet, then the risk related to the sequence $a.b$ is higher than the one of $b.a$.
 - Policy changes: Data could be considered as top secret at a given period of time, but it could be reclassified as public later and the impact of reading it maliciously changes over time.

In the remaining part of this paper, we focus exclusively on action repetition.

The impact of accepting and refusing repeated actions: These information are usually provided by the company depending on the importance of the asset and the nature of the action (read, write, etc.). In other words, we assume that $U_{AH}^c(n \times a)$, $U_{AM}^c(n \times a)$, $U_{DH}^c(n \times a)$ and $U_{DM}^c(n \times a)$ are given. Based on this information, we define the impact on a complete trace as shown by the following definition.

Definition 5. ($U^c(a/\tau^+, \tau^-)$). Let τ^+ be the sequence of previously accepted actions and τ^- be the sequence of previously denied ones. We extend U^c to a trace as follows:

$$U_{AH}^c(a / \tau^+) = U_{AH}^c(|\tau_{\downarrow a}^+| a) + U_{AH}^c(a / |\tau_{\downarrow -a}^+|)$$

$$U_{AH}^c(a / \epsilon) = 0$$

$$U_{AM}^c(a / \tau^+) = U_{AM}^c(|\tau_{\downarrow a}^+| a) + U_{AM}^c(a / |\tau_{\downarrow -a}^+|)$$

$$U_{AM}^c(a / \epsilon) = 0$$

$$U_{DH}^c(a / \tau^-) = U_{DH}^c(|\tau_{\downarrow a}^-| a) + U_{DH}^c(a / |\tau_{\downarrow -a}^-|)$$

$$U_{DH}^c(a / \epsilon) = 0$$

$$U_{DM}^c(a / \tau^-) = U_{DM}^c(|\tau_{\downarrow a}^-| a) + U_{DM}^c(a / |\tau_{\downarrow -a}^-|)$$

$$U_{DM}^c(a / \epsilon) = 0$$

Recall that $|\tau_{\downarrow a}|$ returns the number of actions a in τ and $\tau_{\downarrow -a}$ is the trace τ from which we removed the action a .

We extend the definition of U_{DM}^c to a set of actions A as follows:

$$U_X^c(A / \tau) = \bigcup_{a \in A} U_X^c(a / \tau)$$

Where $X \in \{AH, AM, DH, DM\}$

6.1.2. Hacker Side

When subjects perform malicious actions, they could be detected and risk penalties or not detected and obtain some personal profits. Also, when the user behaves honestly, they could obtain some positive rewards.

For every action and every security property, we need to provide a reward table as shown in Table 6. For the sake of simplicity, we assume, in the remaining part of this paper, that the reward tables are the same for all the subjects.

Table 6. Payoff matrix: user reward for accepted or denied actions.

	Accept (A)	Deny (D)
Honest (H)	U_{AH}^u	U_{DH}^u
Malicious (M)	U_{AM}^u	U_{DM}^u

The reward of repeated actions: If the same action is repeated n times, the reward is not necessarily multiplied by n . In particular, the penalty of doing the same action maliciously could increase exponentially.

This kind of information is provided by the company or by law. Accordingly, we assume that $U_{AH}^u(n \times a)$, $U_{AM}^u(n \times a)$, $U_{DH}^u(n \times a)$ and $U_{DM}^u(n \times a)$ are given. Based on this information, we define the impact on a complete trace as shown by the following definition.

Definition 6. ($U^u(a/\tau^+, \tau^-)$). Let τ^+ be the sequence of previously accepted actions and τ^- be the sequence of previously denied ones. We extend U^u to a trace as follows:

$$U_{AH}^u(a/\tau^+) = U_{AH}^u(|\tau_{\downarrow a}^+|a) + U_{AH}^u(a/|\tau_{\downarrow -a}^+|)$$

$$U_{AH}^u(a/\epsilon) = 0$$

$$U_{AM}^u(a/\tau^+) = U_{AM}^u(|\tau_{\downarrow a}^+|a) + U_{AM}^u(a/|\tau_{\downarrow -a}^+|)$$

$$U_{AM}^u(a/\epsilon) = 0$$

$$U_{DH}^u(a/\tau^-) = U_{DH}^u(|\tau_{\downarrow a}^-|a) + U_{DH}^u(a/|\tau_{\downarrow -a}^-|)$$

$$U_{DH}^u(a/\epsilon) = 0$$

$$U_{DM}^u(a/\tau^-) = U_{DM}^u(|\tau_{\downarrow a}^-|a) + U_{DM}^u(a/|\tau_{\downarrow -a}^-|)$$

$$U_{DM}^u(a/\epsilon) = 0$$

We extend the definition of U_{DM}^u to a set of actions A as follows:

$$U_X^u(A/\tau) = \bigcup_{a \in A} U_X^u(a/\tau)$$

Where $X \in \{AH, AM, DH, DM\}$

6.2. Example of Payoffs

To be able to compute the different impacts on a company as well as the benefits and the penalties of an insider intruder, the **RAFIA** approach requires the following inputs:

- The action versus impact table for every action and every security property.
- The impact of an action if it is repeated many times in a trace, i.e. **impact_A($n \times a$)** and **impact_D($n \times a$)**.
- The action versus reward table for every action and every security property.
- The benefit and the penalty of repeating an action many times in a trace, i.e. **Benefit($s, n \times a$)** and **Penalty($s, n \times a$)**.

The risk associated with a trace is computed by combining the probability that a subject behaves maliciously with the cumulative impact generated by the actions contained in the trace. Let $\rho(s)$ denote the probability that subject s behaves maliciously and let $Impact(\tau)$ represent the cumulative impact derived from the utility functions defined in Definitions 4–6.

$$Risk(\tau) = \rho(s) \times Impact(\tau), \quad (1)$$

where $Impact(\tau)$ aggregates the effects of the actions in τ on the confidentiality, integrity, and availability properties. The resulting risk value is subsequently used by the game-theoretic access-control decision process described in Section 7.

7. Multi-Stage Game

Based on this foundation, the problem can be formalized as a multi-stage game [21]: a sequence of games, each with a different payoff. We also assume that each game is played in a distinct time period. In other words, in a multi-stage game, the history (i.e., the outcome of the previous game) can influence the strategy adopted in the current game.

The system holds two traces: one containing the history of accepted actions and the other containing the history of refused actions. Once a new action is received, there is a risk of adding it to the list of accepted actions and there is a risk of adding it to the list of refused ones. Typically, if the risk of accepting the action is lower than the risk of refusing it, the action is accepted; otherwise, it is refused.

Fig. 7. summarizes the proposed multi-stage game where the game evolves over stages (Game 1, Game 2, ..., Game n), with decisions at each stage affecting future payoffs. Each stage i corresponds to processing action a_i and two traces are maintained: one for the sequence of accepted actions τ^+ and one for the sequence of refused actions τ^- with initial condition: $\tau_0^+ = \tau_0^- = \epsilon$. As previously outlined, the game consists of two players: the company and the user. The company makes a decision to either accept an action with probability p or deny it with probability $1 - p$. The user, on the other hand, can be one of two types: honest or malicious. The user behaves honestly with probability q and acts maliciously with probability $1 - q$. Payoffs represent the utilities or outcomes received by both the company and the user, depending on the actions taken and the type of user involved. These utilities quantify the benefit or cost each player experiences as a result of the interaction.

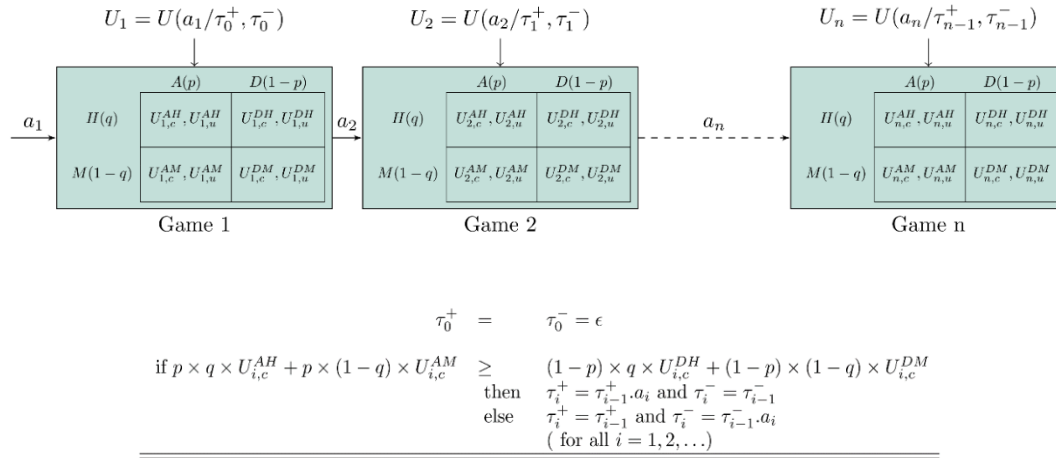


Fig. 7. Multi-stage Game.

To solve the multi-stage game, we use *Nash Equilibrium*, a core concept in game theory that ensures decisions are strategically stable and bridges the gap between theoretical payoffs and real-world actions, transforming abstract risks into probabilities that guide when to accept and when to deny [22]. By combining Nash Equilibrium with updating traces, the model creates a dynamic approach to making decisions resistant to manipulation and rewarding for honest behavior. Nash Equilibrium provides the parameters p and q to quantify risk. The traces τ^+/τ^- dynamically update based on equilibrium strategies.

A Nash Equilibrium (NE) is a pair of strategies (one for the company, one for the user) such that no player can improve their expected utility by unilaterally deviating from their strategy.

- The company decides whether to Accept or Deny based on expected utility.
- The user (or attacker)'s action is honest/malicious.

Hence, the Nash equilibrium simplifies into finding the optimal policy for the company that maximizes expected utility given the action's type probabilities and past history (traces).

The system calculates expected utility:

$$EU_{accept} = p \cdot q \cdot U_{i,cAH} + p \cdot (1-q) \cdot U_{i,cAM} \quad (13)$$

$$EU_{deny} = (1-p) \cdot q \cdot U_{i,cDH} + (1-p) \cdot (1-q) \cdot U_{i,cDM} \quad (14)$$

The company chooses p (probability of Accept) to maximize expected utility:

$$\begin{aligned}
 EU_i &= p[qU_{i,c}^{AH} + (1-q)U_{i,c}^{AM}] \\
 &+ (1-p)[qU_{i,c}^{DH} + (1-q)U_{i,c}^{DM}]
 \end{aligned} \tag{15}$$

First-order condition via derivative:

$$\begin{aligned}
 \frac{dEU_i}{dp} &= qU_{i,c}^{AH} + (1-q)U_{i,c}^{AM} \\
 &- qU_{i,c}^{DH} - (1-q)U_{i,c}^{DM} = 0
 \end{aligned} \tag{16}$$

Solving for the threshold q^* :

$$q^* = \frac{U_{i,c}^{DM} - U_{i,c}^{AM}}{(U_{i,c}^{AH} - U_{i,c}^{DH}) + (U_{i,c}^{DM} - U_{i,c}^{AM})} \tag{17}$$

Company's optimal strategy:

- Accept all if $q > q^*$
- Deny all if $q < q^*$
- Randomize if $q = q^*$

User's Best Response: Malicious users attack when:

$$pU_{i,u}^{AM} + (1-p)U_{i,u}^{DM} \geq 0 \tag{18}$$

In equilibrium, the company makes them indifferent:

$$p^* = \frac{U_{i,u}^{DM}}{U_{i,u}^{DM} - U_{i,u}^{AM}} \tag{19}$$

Malicious users' strategy:

- Attack if acceptance probability $p > p^*$
- Do not attack if $p < p^*$

Multi-stage Dynamics: History updates follow:

$$\begin{aligned}
 \tau_i^+ &= \begin{cases} \tau_{i-1}^+ \cup \{a_i\} & \text{if accepted} \\ \tau_{i-1}^+ & \text{otherwise} \end{cases} \\
 \tau_i^- &= \begin{cases} \tau_{i-1}^- \cup \{a_i\} & \text{if denied} \\ \tau_{i-1}^- & \text{otherwise} \end{cases}
 \end{aligned}$$

Updating:

$$\begin{aligned}
 q_{i+1} &= P(H|a_i, \tau_i) \\
 &= \frac{q_i \cdot P(a_i|H)}{q_i \cdot P(a_i|H) + (1-q_i) \cdot P(a_i|M)}
 \end{aligned}$$

Example 1. Based on the example shown in Table 7 consider the sub-games:

Sub-game 1: Confidentiality $\rightarrow a_1(o_1)$

Payoff Matrix

Table 7. Payoff matrix for Sub-game 1.

Company \ User	Accept (A)	Deny (D)
Honest (H)	(70, 30)	(10, 50)
Malicious (M)	(-10, 90)	(0, 10)

Equilibrium

Company chooses q to make User indifferent:

$$70q - 10 + 10q = 10q \Rightarrow q = \frac{1}{6} \approx 0.167 \quad (20)$$

User chooses p to make Company indifferent:

$$30p + 50 - 50p = 90p + 10 - 10p \Rightarrow p = 0.4 \quad (21)$$

Decision Rule

$$0.4 \cdot 0.167 \cdot 70 + 0.4 \cdot 0.833 \cdot (-10) \geq 0.6 \cdot 0.167 \cdot 10 + 0.6 \cdot 0.833 \cdot 0 \quad (22)$$

$$1.334 \geq 1.002 \Rightarrow \text{Accept } a_1(o_1)$$

Bayesian Update

$$q_2 = \frac{0.167 \cdot 0.8}{0.167 \cdot 0.8 + 0.833 \cdot 0.2} \approx 0.45 \quad (23)$$

Sub-game 2: Confidentiality $\rightarrow a_2(o_1)$

Payoff Matrix

The payoff matrix for Sub-game 2 is presented in Table 8

Table 8. Payoff matrix for Sub-game 2.

Company \ User	Accept (A)	Deny (D)
Honest (H)	(60, 40)	(15, 55)
Malicious (M)	(-15, 85)	(5, 20)

Equilibrium

$$60q - 15 + 15q = 15q + 5 - 5q \Rightarrow 65q = 20 \Rightarrow q \approx 0.308 \quad (24)$$

Company chooses q to make User indifferent:

User chooses p to make Company indifferent:

$$40p + 55 - 55p = 85p + 20 - 20p \Rightarrow 100p = 35 \Rightarrow p = 0.35 \quad (25)$$

Decision Rule

$$EU_{Accept} \approx 0.35 \cdot 0.308 \cdot 60 + 0.35 \cdot 0.692 \cdot (-15) = -1.46 \quad (26)$$

$$EU_{Deny} \approx 0.65 \cdot 0.308 \cdot 15 + 0.65 \cdot 0.692 \cdot 5 = 5.25 \quad (27)$$

Decision: Deny $a_2(o_1)$

Bayesian Update

$$q_3 = \frac{0.45 \cdot 0.7}{0.45 \cdot 0.7 + 0.55 \cdot 0.3} \approx 0.656 \quad (28)$$

Based on this analysis, Table 9 summarizes a sequence of sub-games that illustrate how game-theoretic principles, particularly the Nash Equilibrium—are employed to inform decisions in uncertain environments. Each case represents an encounter between a company and a user, whose behavior could be either trustworthy or harmful. The table illustrates the idea of how Nash Equilibrium completes the decision-making model by establishing rational thresholds: it demonstrates that the malicious user's optimal strategy is determined by the equilibrium probability p^* , while the company's optimal acceptance strategy relies on the trust threshold q^* . These threshold values represent equilibrium points where neither party has a reason to change their strategy independently. Also, the table illustrates how Nash Equilibrium accommodates mixed strategies by allowing probabilistic decision-making in situations of uncertainty, reflecting a realistic and adaptive approach. Moreover, the system enhances stability through Bayesian trust updates, which refine the estimation of a user's honesty (q) based on their previous interactions. As a result, Nash Equilibrium effectively closes the strategic loop between the two players—balancing the company's risk tolerance with the user's incentive structure—and translates abstract utility trade-offs into actionable thresholds (e.g., $q^* = 0.167$). This framework provides a principled basis for risk-aware access-control decisions that evolve over time.

This equilibrium plays a critical role in ensuring that both parties reach a stable strategic point—where neither has an incentive to change their behavior unilaterally—thus enabling more secure, adaptive, and utility-maximizing access control decisions over time.

Table 9. Multi-stage Sub-game Analysis

Sub-game	Action	Security Prop.	Expected Utility	p	q	Decision	Updated q	Update Rule
			Accept Deny	Nash p^*	Trust q			
1	$a_1(o_1)$	Confidentiality	1.87 1.0	0.40	0.167	Accept	0.45	$\frac{0.167 \cdot 0.8}{0.167 \cdot 0.8 + 0.833 \cdot 0.2} \approx 0.45$
2	$a_2(o_1)$	Integrity	-1.46 5.25	0.35	0.45	Deny	0.656	$\frac{0.45 \cdot 0.7}{0.45 \cdot 0.7 + 0.55 \cdot 0.3} \approx 0.656$
3	$a_3(o_1)$	Availability	0.24 1.30	0.25	0.656	Deny	0.816	$\frac{0.656 \cdot 0.7}{0.656 \cdot 0.7 + 0.344 \cdot 0.3} \approx 0.816$
4	$a_1(o_1)$	Integrity	52.12 18.16	0.60	0.816	Accept	0.922	$\frac{0.816 \cdot 0.8}{0.816 \cdot 0.8 + 0.184 \cdot 0.2} \approx 0.922$
5	$a_2(o_1)$	Integrity	63.76 24.22	0.75	0.922	Accept	0.947	$\frac{0.922 \cdot 0.6}{0.922 \cdot 0.6 + 0.078 \cdot 0.4} \approx 0.947$

Example 2. Fig. 8 provides an example of the information required to evaluate impacts, benefits, and penalties. In this table ι is used to represent impacts when referring to the company, and it denotes benefits and rewards when referring to the attacker. In addition, we assume that the same formulas apply to both **impact_A** and **impact_D**, as well as for **Benefit** ($s, n \times a$) and **Penalty** ($s, n \times a$). For example, if we write $\iota(n \times a_1(o_1)) = n \times \iota(a_1(o_1))$ in the context of company impacts, this means that **impact_A**($n \times a_1(o_1)$) = $n \times \text{impact}_A(a_1(o_1))$ and **impact_D**($n \times a_1(o_1)$) = $n \times \text{impact}_D(a_1(o_1))$. Table 10 shows how to evaluate the payoffs for repeated actions.

Table 10. Example of the payoffs of actions a1, a2 and a3 on objects o1 and o2 repeated n times

		$U(n \times a_1(o_1))$	$U(n \times a_2(o_1))$	$U(n \times a_3(o_1))$	$U(n \times a_1(o_2))$	$U(n \times a_2(o_2))$	$U(n \times a_3(o_2))$
Company	Confidentiality	$U(a_1(o_1))$	$U(a_2(o_1))$	$U(a_3(o_1))$	$U(a_1(o_2))$	$U(a_2(o_2))$	$U(a_3(o_2))$
	Integrity	$U(a_1(o_1))$	$U(a_2(o_1))$	$U(a_3(o_1))$	$U(a_1(o_2))$	$U(a_2(o_2))$	$U(a_3(o_2))$
	Availability	$n \times U(a_1(o_1))$	$n \times U(a_2(o_1))$	$n \times U(a_3(o_1))$	$n \times U(a_1(o_2))$	$n \times U(a_2(o_2))$	$n \times U(a_3(o_2))$
User payoffs	Confidentiality	$U(a_1(o_1))$	$U(a_2(o_1))$	$U(a_3(o_1))$	$U(a_1(o_2))$	$U(a_2(o_2))$	$U(a_3(o_2))$
	Integrity	$U(a_1(o_1))$	$U(a_2(o_1))$	$U(a_3(o_1))$	$U(a_1(o_2))$	$U(a_2(o_2))$	$U(a_3(o_2))$
	Availability	$n \times U(a_1(o_1))$	$n \times U(a_2(o_1))$	$n \times U(a_3(o_1))$	$n \times U(a_1(o_2))$	$n \times U(a_2(o_2))$	$n \times U(a_3(o_2))$

		$a_1(o_1)$	$a_2(o_1)$	$a_3(o_1)$	$a_1(o_2)$	$a_2(o_2)$	$a_3(o_2)$
Company payoffs	Confidentiality	$\begin{matrix} & A & D \\ H & (70, 30) & (10, 50) \\ M & (-10, 90) & (0, 10) \end{matrix}$	$\begin{matrix} & A & D \\ H & (60, 40) & (15, 55) \\ M & (-15, 85) & (5, 20) \end{matrix}$	$\begin{matrix} & A & D \\ H & (50, 20) & (5, 30) \\ M & (-20, 100) & (0, 15) \end{matrix}$	$\begin{matrix} & A & D \\ H & (55, 35) & (12, 45) \\ M & (-12, 80) & (3, 18) \end{matrix}$	$\begin{matrix} & A & D \\ H & (45, 25) & (8, 35) \\ M & (-25, 95) & (2, 12) \end{matrix}$	$\begin{matrix} & A & D \\ H & (35, 15) & (6, 25) \\ M & (-35, 105) & (1, 8) \end{matrix}$
	Integrity	$\begin{matrix} & A & D \\ H & (65, 50) & (20, 60) \\ M & (-5, 70) & (10, 30) \end{matrix}$	$\begin{matrix} & A & D \\ H & (55, 45) & (18, 55) \\ M & (-8, 75) & (7, 25) \end{matrix}$	$\begin{matrix} & A & D \\ H & (75, 55) & (22, 65) \\ M & (-3, 65) & (12, 35) \end{matrix}$	$\begin{matrix} & A & D \\ H & (60, 40) & (15, 50) \\ M & (-10, 80) & (5, 20) \end{matrix}$	$\begin{matrix} & A & D \\ H & (50, 30) & (10, 40) \\ M & (-20, 90) & (0, 10) \end{matrix}$	$\begin{matrix} & A & D \\ H & (80, 60) & (25, 70) \\ M & (0, 60) & (15, 40) \end{matrix}$
	Availability	$\begin{matrix} & A & D \\ H & (30, 10) & (5, 20) \\ M & (-40, 110) & (-5, 5) \end{matrix}$	$\begin{matrix} & A & D \\ H & (25, 5) & (3, 15) \\ M & (-45, 115) & (-8, 3) \end{matrix}$	$\begin{matrix} & A & D \\ H & (20, 0) & (2, 10) \\ M & (-50, 120) & (-10, 0) \end{matrix}$	$\begin{matrix} & A & D \\ H & (35, 15) & (7, 25) \\ M & (-35, 105) & (0, 10) \end{matrix}$	$\begin{matrix} & A & D \\ H & (200, 20) & (10, 0) \\ M & (10, 100) & (0, 10) \end{matrix}$	$\begin{matrix} & A & D \\ H & (0, 0) & (0, 0) \\ M & (0, 0) & (0, 0) \end{matrix}$
	Availability						
User payoffs	Confidentiality	$\begin{matrix} & A & D \\ H & (30, 70) & (50, 10) \\ M & (90, -10) & (10, 0) \end{matrix}$	$\begin{matrix} & A & D \\ H & (40, 60) & (55, 15) \\ M & (85, -15) & (20, 5) \end{matrix}$	$\begin{matrix} & A & D \\ H & (20, 50) & (30, 5) \\ M & (100, -20) & (15, 0) \end{matrix}$	$\begin{matrix} & A & D \\ H & (35, 55) & (45, 12) \\ M & (80, -12) & (18, 3) \end{matrix}$	$\begin{matrix} & A & D \\ H & (25, 45) & (35, 8) \\ M & (95, -25) & (12, 2) \end{matrix}$	$\begin{matrix} & A & D \\ H & (15, 35) & (25, 6) \\ M & (105, -35) & (8, 1) \end{matrix}$
	Integrity	$\begin{matrix} & A & D \\ H & (50, 65) & (60, 20) \\ M & (70, -5) & (30, 10) \end{matrix}$	$\begin{matrix} & A & D \\ H & (45, 55) & (55, 18) \\ M & (75, -8) & (25, 7) \end{matrix}$	$\begin{matrix} & A & D \\ H & (55, 75) & (65, 22) \\ M & (65, -3) & (35, 12) \end{matrix}$	$\begin{matrix} & A & D \\ H & (40, 60) & (50, 15) \\ M & (80, -10) & (20, 5) \end{matrix}$	$\begin{matrix} & A & D \\ H & (30, 50) & (40, 10) \\ M & (90, -20) & (10, 0) \end{matrix}$	$\begin{matrix} & A & D \\ H & (60, 80) & (70, 25) \\ M & (60, 0) & (40, 15) \end{matrix}$
	Availability	$\begin{matrix} & A & D \\ H & (10, 30) & (20, 5) \\ M & (110, -40) & (5, -5) \end{matrix}$	$\begin{matrix} & A & D \\ H & (5, 25) & (15, 3) \\ M & (115, -45) & (3, -8) \end{matrix}$	$\begin{matrix} & A & D \\ H & (0, 20) & (10, 2) \\ M & (120, -50) & (0, -10) \end{matrix}$	$\begin{matrix} & A & D \\ H & (15, 35) & (25, 7) \\ M & (105, -35) & (10, 0) \end{matrix}$	$\begin{matrix} & A & D \\ H & (20, 200) & (0, 10) \\ M & (100, 10) & (10, 0) \end{matrix}$	$\begin{matrix} & A & D \\ H & (0, 0) & (0, 0) \\ M & (0, 0) & (0, 0) \end{matrix}$
	Availability						

 Fig. 8. Example of the payoffs of actions a_1 , a_2 and a_3 on objects o_1 and o_2 .

8. Risk-LTL: The Proposed Language to Specify Security Policies within RAFIA

The RAFIA framework relies on security policies specified in Risk-LTL. While the proposed risk-evaluation model and game-theoretic decision process can be understood independently of the policy language, Risk-LTL provides the formal mechanism used to specify security requirements and risk-aware access-control constraints. This section presents the syntax and semantics of Risk-LTL and explains how it supports the access-control decisions described in the previous sections. Many organizations use classical models such as RBAC or ABAC to specify their security policies. RBAC, for example, has dominated access control for nearly a quarter of a century [16,23]. Within RBAC, a security manager defines roles and assigns permissions to them. The role can include more than one user and a user can be assigned to more than one role [16,24]. Nevertheless, even though these access models provide good protection against outsider attacks, they do not offer any defense against insider attacks. For example, they cannot mitigate the risk that an employee with access to critical data might provide it to a competing company. Therefore, a more appropriate language for specifying security policies should also take into consideration the risks of insider attacks, along with other features such as:

- The repetition of actions: performing the same action many times can increase the risk. For example, if an employee reads the record of one client, the risk may be low; however, if one million records are read by the same employee, the risk may become high.
- The order between actions: if an action a is performed before b , the risk would be acceptable, but the reverse might not be true. For example, having a connection to the Internet and then accessing sensitive information is usually less risky than the reverse. In fact, once an employee gains access to sensitive data and then connects to the Internet, the data can be sent to the outside.
- The conflict of interest: If an action a is performed by an employee A and the action b by an employee B , the risk may be low. However, if they are done by the same employee, the risk could be high (it could involve a conflict of interest for example).

The objective is to specify, detect, and prevent such security-critical situations. We believe that a language like Linear Temporal Logic (LTL), endowed with risk-awareness and called Risk-LTL, will be suitable for enhancing security models such as ABAC or RBAC with the capacity to prevent insider attacks.

8.1. Syntax of Risk-LTL

Let $\Sigma = \langle S, O, A, C \rangle$ be an **ACD**. Let $S \in 2^S$, $A \in 2^A$, $O \in 2^O$ and $C \in 2^C$. The syntax of an access control policy based on Risk-LTL is as follows:

$$\begin{aligned}
 \Phi, \Psi &::= p \mid \neg\Phi \mid \Phi \vee \Psi \mid X\Phi \mid \Phi U \Psi \mid \text{Risk}_*\lambda(\Phi) \\
 p &::= (S, A, O, C) \\
 * &\in \{<, \leq, >, \geq\} \\
 \lambda &\in \mathbb{R}
 \end{aligned}$$

A formula in Risk-LTL can be a simple proposition p , the negation of a formula ($\neg\Phi$), the disjunction of two formulas ($\Phi \vee \Psi$), the next of a formula ($X\Phi$), a formula Φ until Ψ ($\Phi U \Psi$) and a formula with a threshold value λ of risk

$(Risk_{\star\lambda}(\Phi))$. A proposition p can specify a set of actions at the same time by specifying a set of users S , a set of action names A , a set of objects O and a set of contexts C .

8.2. Semantics of Risk-LTL

Intuitively, the semantics of the Risk-LTL formulas are as follows:

- p : a trace τ respects p if its first action is in p .
- $\neg\Phi$: a trace τ respects $\neg\Phi$ if it does not respect Φ .
- $\Phi \vee \Psi$: a trace τ respects $\Phi \vee \Psi$ if it respects Φ or it respects Ψ .
- $X\Phi$: a trace τ respects $X\Phi$ if τ^1 respects Φ .
- $\Phi U \Psi$: a trace τ respects $\Phi U \Psi$ if there exists $i, i \geq 0$, such that τ^i respects Ψ and for all $j, j < i, \tau^j$ respects Φ .
- $Risk(\Phi)$: a trace τ respects $Risk(\Phi)$ if it respects Φ and the risk of accepting it is lower than the risk of denying it.

More formally, let $\mathcal{R}$ be an RM and $\langle \tau^+, \tau^- \rangle$ a trace containing the history of accepted and refused action respectively. We write $\langle \tau^+, \tau^- \rangle \models_{\mathcal{R}} \Phi$ to say that the trace τ respects the formula Φ and the risk is acceptable. We also use $\tau \vdash \Phi$ to say that τ respects Φ without taking any risk into consideration.

$$\begin{array}{ll}
 \tau \vdash p & \text{if } \tau(0) \sqsubseteq p \\
 \tau \vdash \neg\Phi & \text{if } \tau \not\vdash \Phi \\
 \tau \vdash \Phi \vee \Psi & \text{if } \tau \vdash \Phi \text{ or } \tau \vdash \Psi \\
 \tau \vdash X\Phi & \text{if } \tau^1 \vdash \Phi \\
 \tau \vdash \Phi U \Psi & \text{if } \exists i, i \geq 0 \mid \tau^i \vdash \Psi \\
 & \text{and } \forall j, j < i, \tau^j \vdash \Phi
 \end{array}$$

$$\langle \tau^+, \tau^- \rangle \models_{\mathcal{R}} Risk(\Phi)$$

$$\text{if } \tau^+ \vdash \Phi$$

$$\text{and } \mathcal{R}(\tau^+) \geq \mathcal{R}(\tau^-).$$

where

$$a(s, o, c) \sqsubseteq (S, A, O, C) \text{ if } (s, a, o, c) \in S \times A \times O \times C.$$

8.3. Abbreviations

We can add some abbreviations (macros) to Risk-LTL such as:

$$\begin{array}{ll}
 true & \equiv p \vee \neg \\
 false & \equiv p \neg true \\
 \Phi \wedge \Psi & \equiv \neg(\neg\Phi \vee \neg\Psi) \\
 \Phi \rightarrow \Psi & \equiv \Psi \vee \neg\Phi \\
 \Phi \leftrightarrow \Psi & \equiv (\Phi \rightarrow \Psi) \wedge (\Psi \rightarrow \Phi)
 \end{array}$$

Finally (F): Φ should be true in at least one future state of the trace:

$$F(\Phi) \equiv true U \Phi$$

Globally (G): Φ should be true in every future state of the trace:

$$G(\Phi) \equiv \neg F(\neg\Phi)$$

9. Related Works

This section reviews the most relevant literature related to risk-aware access control and techniques for mitigating insider attacks. Risk-based access control has been extensively investigated during the past two decades and remains an active research area because of the growing need for adaptive authorization mechanisms capable of balancing security

and usability. Consequently, numerous quantitative and qualitative risk-estimation techniques have been proposed to support authorization decisions in dynamic environments. The following discussion reviews the most relevant approaches and highlights their strengths and limitations.

Ref. [9] developed a model using fuzzy logic to measure the risks associated with each access request, aiming to support access control decisions. The motivation behind this work is to address the challenge of enabling organizations to share sensitive data while supporting rapid decision-making. The model calculates risk based on the value of information and the probability of unauthorized disclosure, using subject trustworthiness and object sensitivity as key risk factors. It measures the difference between these factors to determine the risk value associated with each access request. The greater the difference, the higher the risk, and access is consequently denied; conversely, a smaller difference results in a lower risk and access is permitted. Nevertheless, the paper does not provide sufficient details regarding the implementation of the risk-quantification process.

Ref. [25] proposed a dynamic risk-based access control model. The approach relies on risk policies that allow users and service providers to define how risk-based access to their resources should be handled, employing various quantification and aggregation methods. This design provides high flexibility to the access control system. However, the model requires system administrators to ensure that a minimum level of security is maintained, which limits its effectiveness in addressing insider attacks.

Ref. [26] developed a mechanism to protect cloud environments from insider attacks by combining trust evaluation with risk-based access control. Their approach integrates a keystroke analysis mechanism into the access control model to detect harmful insider behavior by classifying user activity based on keystroke content. Information from both the keystroke analyzer and the risk analyzer is used to compute a threshold that determines whether an access request should be granted or denied. While the approach is innovative, it has limitations in effectively detecting sophisticated insider threats. Ref. [27] proposed an access control mechanism that combines trust and risk to evaluate access decisions. Their evaluation demonstrates that integrating risk assessment with trust (or confidence) improves the detection of malicious attacks. Nevertheless, the model lacks the capability to effectively detect insider threats, which require more advanced behavioral correlation across multiple users.

Ref. [28] proposed a risk-based access-control model for granting access based on an assessment of security risks and operational need. It takes into account various risk identification factors such as user confidence, data sensitivity, access decision history, user roles, etc. The estimated risk value is compared to a threshold provided by the organization's security policy. Then, based on the policies and operational need, a decision is made to grant or deny access. The overall idea of this approach seems sound, but it does not provide a method for estimating the risk value. Ref. [29] proposed a method to make access decisions using risk assessment to estimate the risk to confidentiality, integrity, and availability. The measured risk value associated with each access request is compared with a threshold to determine whether access should be granted or denied. The approach is promising, but the quantitative evaluations are not clearly explained, and it also lacks contextual information as a factor in access-control decision making.

Ref. [13] proposed a framework of game-theoretical risk-based access control. The risk is calculated from the relationship between the subjects and the system. Once the request is received, the risk is evaluated using the user context, resource context, access history, risk history, and game history. Then, the system tries to discover the access strategy of the user who requests the access by identifying whether the request violates the security policy or not. The system performs its optimal strategy according to the user's strategy to get maximal benefits, and updates the game history. The optimal strategy adopted by the system determines the access decision for the received access request. If the privacy risk of the user access request is greater than the threshold value, it will be considered as a privacy violation access; otherwise, the user request is classified as benign. The use of game theory provides a good way to establish the appropriate strategies; however, the paper does not explain how to evaluate the quantitative values of the risk. Similarly, Ref. [30] used game theory in access decisions by proposing a method that estimates the risk value associated with an access request and the value of user benefits. However, it does not explain how to quantitatively compute the risk values.

In contrast to the game-theoretic access-control approaches presented in [13,30], RAFIA explicitly integrates trace-based risk computation, Risk-LTL policy enforcement, and multi-stage game-theoretic reasoning within a unified access-control framework. Consequently, access decisions depend not only on the current request but also on accumulated behavioral history and dynamically evaluated risk.

Ref. [31] proposed a method to make access decisions by using the risk assessment to estimate the risk for confidentiality, integrity and availability. The measured risk value associated with each access request is compared with a threshold to decide whether access should be granted or denied. The approach is promising, but the quantitative evaluations are not clearly explained and it also lacks contextual features.

The comparison presented in Table 11 is based on the primary security objectives, risk-computation factors, and methodological characteristics explicitly discussed in the corresponding publications. This comparison also includes recent graph-based, deep-learning-based, and AI-assisted approaches relevant to insider-threat detection and access-control decision making. Another method was proposed by [32] for determining whether to grant or deny access based on a comparison of risk values and user trust level related to the request. The higher the user's trust level compared to the risk values, the access is granted; otherwise, it will be denied. Using the notion of risk to grant access is the main advantage of the approach; however, it is ineffectual in preventing insider attacks.

Ref. [33] proposed a comprehensive framework for risk assessment in access control systems by evaluating and integrating four distinct risk estimation methods. These methods include assessing risk based on subject trustworthiness, object sensitivity, the difference between the two, and a hybrid approach that combines multiple factors. The estimated value of risk for an access request varies according to the threat assessment approach that is chosen. Accordingly, the risk-assessment function must be carefully selected according to the application context and the security requirements of the organization. However, the quantitative formulation of risk is not clearly specified. In addition, the approach does not include contextual information as a factor in access-control decision making.

Ref. [34] developed a dynamic access control model for e-Health cloud systems using risk that ensures access granting based on confidentiality, integrity, and availability requirements. The model estimates the risk related to user actions by using a mathematical function. Also, the system looks into the previous users' behavior patterns to calculate the final risk score and compares it with a predetermined risk threshold. If the risk score is lower than this threshold, access is granted; otherwise, access is denied.

Ref. [35] used fuzzy logic along with expert judgment to estimate security risks based on subject context, object sensitivity, and action severity. The evaluated risk value is then compared to a predefined risk threshold to support decision-making. This model was specifically proposed for IoT environments.

Ref. [36] proposed a risk-aware information disclosure model that evaluates the risk based on the contextual information surrounding the data request. The intention behind this is to address the problem of sharing biomedical data and controlling access to them. Biomedical data are typically highly sensitive and the impact of an unauthorized disclosure is high. Ref. [37] proposed a model for assessing the risks associated with access requests to medical data. The goal is to prevent insider threats, such as misuse and unauthorized disclosure of health records. Risk assessment in the model is based on the user's confidence and the sensitivity of the data object. The resulting score is compared to a threshold to decide whether to grant or deny the access request. However, the model still struggles to detect complex insider threats.

Ref. [38] proposed a technique to prevent insider attacks related to cloud services and data. A particular challenge in cloud computing is that the service provider itself may act as an insider attacker. Nevertheless, detecting or preventing attacks caused by a service provider could be a very sophisticated task or even impossible.

Ref. [39] provided an approach to protect databases from insider attacks by triggering alarms when any access control violation is detected. The proposed model evaluates risks arising from access requests and classifies them according to a threshold value.

Ref. [40] proposed a risk-based access control framework specifically designed for Hospital Information Systems (HIS), to improve decision-making by accounting for dynamic security conditions. Rather than relying on fixed access rules, the proposed model adapts permissions based on real-time assessments. These assessments consider factors such as the user's role, the type of action requested, and the sensitivity of the data involved. While the approach is well-positioned to handle insider threats in healthcare settings (where patient confidentiality is critical), it currently lacks a clear methodology for computing risk scores in practice. The model is tailored to healthcare environments, which may limit its applicability to other domains with different operational requirements. It also does not incorporate mechanisms for measuring trust or tracking the evolution of user behavior over time, features that are becoming increasingly important in today's systems for spotting insider risks and adapting accordingly.

Ref. [41] proposed Deep Learning Based Access Control (DLBAC), an approach that applies deep neural networks to automate access-control decisions using user and resource metadata. Instead of relying on manually engineered authorization policies, the model learns access-control rules directly from historical authorization data and organizational information. The approach was evaluated using both synthetic and real-world datasets and demonstrated promising results in terms of authorization accuracy and generalization capabilities. The use of deep learning reduces the complexity of policy engineering and supports adaptive access-control decisions. However, the approach focuses primarily on learning authorization behavior from data and does not explicitly incorporate quantitative risk estimation, temporal policy specification, trace-based risk evaluation, or game-theoretic reasoning. Consequently, it does not provide a mechanism for evaluating insider-attack risk over execution traces before authorization decisions are made.

Ref. [42] proposed a heterogeneous graph-based semi-supervised learning framework for access-control decision making. The approach constructs an access-control heterogeneous graph that models users, resources, departments, managers, and organizational relationships, and employs graph neural networks to learn node embeddings that support authorization decisions. Experimental evaluation on the Amazon access-control dataset demonstrated improved decision-making performance compared with traditional feature-based approaches. While the approach effectively exploits organizational relationships and machine-learning techniques, its primary objective is authorization prediction rather than risk-aware insider-threat mitigation.

Ref. [43] proposed an adaptive access-control architecture for protecting privacy-sensitive healthcare data using Large Language Models (LLMs). The approach combines context-aware risk assessment, anomaly detection, query sanitization, and dynamic threat modeling to secure electronic health records and ensure compliance with privacy regulations such as HIPAA and GDPR. Access-control decisions are continuously adapted according to the estimated risk associated with user activities and detected threats. Experimental results indicate improvements in threat-detection performance and reductions in false positives compared with traditional security approaches. While the framework provides adaptive risk-aware authorization through artificial intelligence techniques, it does not offer a formal temporal-

logic-based policy specification language or a game-theoretic decision model for analyzing strategic interactions between users and the organization.

Table 11 shows a comparison of some previous works with the proposed approach (RAFIA), which is presented in the last row of the table.

Table 11. Comparison of RAFIA with Existing Risk-Aware Access-Control Approaches.

Work	Protected CIA		Method	Risk Computation		Factors	Scope	
	Confidentiality Integrity	Availability		Subject	Object Action	Trust	Context	
[9]	✓	✓	Fuzzy Logic	✓	✓			IBM Systems
[25]	✓	✓	Mathematical Equation					Cloud
[26]	✓		Risk Assessment			✓		Cloud Computing
[27]	✓		Risk Assessment	✓	✓	✓		Grid Environment
[28]	✓		Risk Assessment	✓	✓			National Security Agency
[29]	✓		Mathematical Equation	✓	✓	✓		Healthcare, Military
[13]	✓		Game Theory	✓	✓			Healthcare
[30]	✓		Game Theory	✓				Online bookstore scenario
[31]	✓	✓	Risk Assessment		✓			Hospital Case Study
[32]	✓		Mathematical Equation			✓		Threat Detection Systems
[33]	✓	✓	Risk Assessment		✓	✓		Organizations
[34]	✓	✓	Mathematical Equation	✓	✓	✓		Cloud Computing/eHealth
[35]	✓	✓	Fuzzy Logic		✓	✓	✓	IoT
[36]	✓		Mathematical Equation		✓		✓	Biomedical data
[37]	✓	✓	Mathematical Equation		✓	✓		Healthcare
[38]	✓	✓	Cryptography	✓				Cloud
[39]	✓	✓	Risk Assessment	✓	✓	✓		DBMS
[40]	✓		Risk Assessment	✓	✓	✓	✓	Healthcare
[41]	✓		Deep Learning	✓	✓	✓	✓	Enterprise Access Control
[42]	✓	✓	Graph Neural Network	✓	✓		✓	Enterprise Organizations
[43]	✓	✓	LLM-Based Access Control	✓	✓	✓	✓	Healthcare
RAFIA	✓	✓	Mathematical Equation	✓	✓	✓	✓	Generalized

As shown in the last column of this table, although all of these approaches tried to estimate the risk associated with access-control decisions, most of them are restricted by the nature of the problem (IoT, Healthcare, etc.) and the environment (Cloud, DBMS, etc.) in which the risk assessment technique is applied, which limits their generalizability. Furthermore, most existing approaches focus primarily on confidentiality and privacy, which are crucial security properties, but few of them deal with integrity and availability, which are also very important. Also, the majority of the approaches do not take into consideration most of the relevant parameters (subject, object, action, trust, context) when estimating the risk value associated with any access request.

Unlike existing risk-aware access-control approaches that typically focus on either risk estimation, policy specification, or strategic decision making independently, RAFIA integrates these aspects within a unified framework. More specifically, RAFIA combines (i) Risk-LTL for expressing risk-aware security policies, (ii) trace-based risk evaluation that incorporates historical user behavior and action repetition, and (iii) a multi-stage game-theoretic model that dynamically adapts access-control decisions according to evolving trust levels and estimated risks. This combination distinguishes RAFIA from existing approaches and provides a comprehensive framework for insider-threat-aware access control.

10. Prototype Implementation and Experimental Evaluation

Access control remains a cornerstone of information security, traditionally relying on static policies. However, static approaches fail to adapt to evolving user behaviors and contextual changes, leading to either excessive denial of legitimate requests or failure to detect malicious activities. This section presents the implementation of the RAFIA prototype and its experimental evaluation. The prototype was developed in Python using the Tkinter framework to demonstrate the proposed game-theoretic risk-aware access-control model. Beyond illustrating the graphical user interface, the prototype was evaluated using an automated benchmarking framework that generates synthetic access-request traces and measures

both detection performance and computational efficiency. The experiments compare RAFIA against a static-threshold access-control baseline using standard classification metrics and implementation performance indicators.

The prototype system is composed of the following main modules:

- **Policy Check:** Validates incoming access requests against predefined policies, ensuring that permissions align with user roles, requested actions, targeted objects, and current context (e.g., physical location).
- **Risk Computation:** Evaluates each request by calculating a risk score and maliciousness probability. This evaluation considers contextual deviations, frequency of similar requests, and behavioral anomalies.
- **Historical Log and Learning:** Maintains a persistent log of accepted and denied requests, enabling the system to learn patterns and adjust risk thresholds dynamically based on observed behaviors.
- **Decision Making:** Combines policy compliance and risk evaluation to accept or deny requests, mitigating potential threats effectively.
- **User Interface:** A graphical user interface built with Python's Tkinter framework allows users and administrators to submit requests, view decisions, risk metrics, and logs transparently.

10.1. Prototype Implementation

The prototype processes each access request according to the following sequence:

1. **Input:** Users provide four parameters via the GUI: *User*, *Action*, *Object*, and *Context* Fig. 9.
2. **Policy Verification:** The system verifies if the request complies with the access control policies.
3. **Risk Evaluation:** A risk score and maliciousness probability are computed using request attributes and historical data.

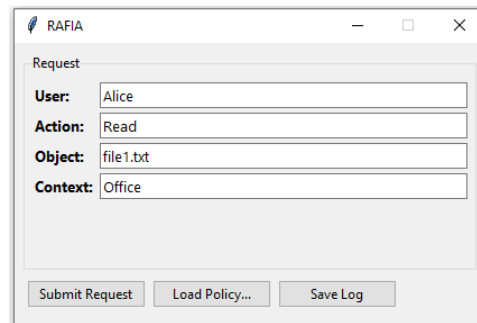


Fig. 9. RAFIA prototype interface for submitting an access request.

Learning: Past execution traces (τ^+ , τ^-) are analyzed to detect anomalies or repeated unusual requests Fig. 10.

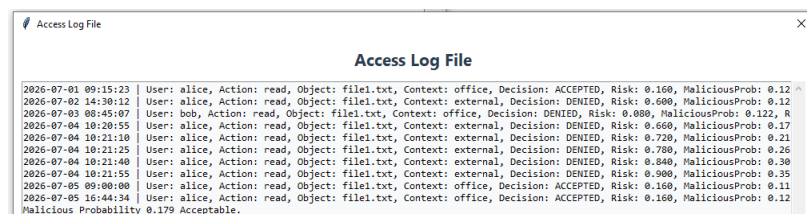


Fig. 10. RAFIA log showing previously accepted and denied access requests.

4. **Decision Output:** The system returns a decision (ACCEPTED or DENIED) along with detailed justifications and risk metrics Fig. 11 and Fig. 12.

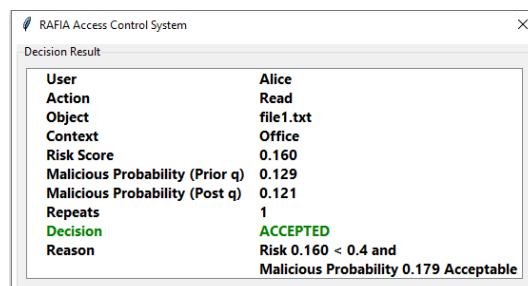
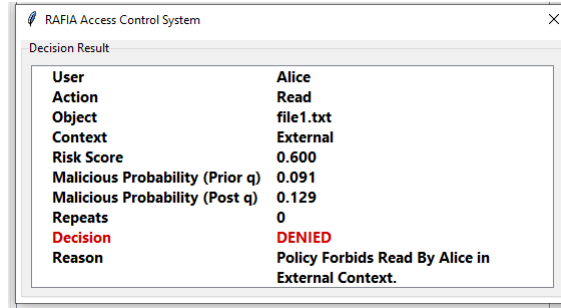


Fig. 11. Decision returned by the RAFIA prototype for a legitimate access request.



The screenshot shows a window titled "RAFIA Access Control System" with a "Decision Result" section. It contains a table with the following data:

User	Alice
Action	Read
Object	file1.txt
Context	External
Risk Score	0.600
Malicious Probability (Prior q)	0.091
Malicious Probability (Post q)	0.129
Repeats	0
Decision	DENIED
Reason	Policy Forbids Read By Alice in External Context.

Fig. 12. Decision returned by the RAFIA prototype for a denied access request due to policy violation and elevated risk.

5. **Logging:** All request data and decisions are saved for continuous learning and audit.

10.2. Adaptive Learning Mechanism

The system incorporates game-theoretic reasoning by treating user interactions as repeated moves in a security game. Historical logs allow modeling of user behavior, enabling risk scores to be dynamically adjusted. For instance, repeated access from trusted contexts (e.g., office) is tolerated, whereas access from unusual contexts (e.g., home or external locations) escalates risk and leads to denial upon threshold breach.

10.3. Prototype Demonstration

The following scenarios illustrate the operation of the implemented RAFIA prototype under representative access-control situations. These examples demonstrate the correctness and functionality of the prototype implementation. Quantitative performance evaluation is presented separately in the following subsection.

- **Scenario 1:** User Alice accesses file1.txt from the office context. The request complies with policies, exhibits low risk, and is accepted.
- **Scenario 2:** Alice attempts access from the home context, triggering increased risk and denial due to unusual location.
- **Scenario 3:** User Bob, unauthorized by policy, attempts to read file1.txt and is denied immediately.
- **Scenario 4:** Repeated access attempts from abnormal contexts result in escalating risk and automatic denial.

These representative scenarios demonstrate that RAFIA correctly combines policy verification, quantitative risk estimation, and historical behavioral information to produce adaptive authorization decisions. Legitimate requests are accepted when the associated risk remains below the configured threshold, whereas policy violations and repeated abnormal behaviors result in automatic denial.

10.4. Experimental Evaluation

10.4.1. Experimental Setup

To quantitatively evaluate RAFIA, we developed an automated benchmarking framework in Python that executes large numbers of synthetic access requests without requiring user interaction through the graphical interface. Unlike the prototype screenshots presented in the previous subsection, which illustrate the functionality of the implementation, the benchmark provides quantitative measurements of both the effectiveness and computational performance of the proposed framework.

The benchmark repeatedly submits access requests to the RAFIA decision engine, records the resulting authorization decisions, and measures execution time, memory consumption, throughput, and detection performance. The evaluation considers four workload sizes (100, 1,000, 5,000, and 10,000 requests), representing small, medium, large, and stress-test deployment scenarios, respectively. Each experiment was repeated ten times to reduce measurement variability, and the reported results correspond to the mean values together with the standard deviations.

The benchmarking framework was implemented as a standalone Python program that automatically generates synthetic access requests, executes the RAFIA decision engine, records the authorization outcomes, and computes all evaluation metrics. The framework also exports the raw experimental data and summary statistics in CSV format, thereby facilitating the reproducibility of the reported experiments.

10.4.2. Experimental Environment

The RAFIA prototype and the benchmarking framework were implemented in Python 3.12 and executed on a workstation running Microsoft Windows 10 Pro (64-bit). The experimental platform was equipped with an Intel® Core™ i7-7820HQ processor operating at 2.90 GHz (4 physical cores and 8 logical processors) and 8 GB of RAM.

Execution time was measured using Python's `time.perf_counter()` function, while peak memory consumption was monitored using the `tracemalloc` library. All experiments were performed on the same hardware configuration to ensure consistent and reproducible measurements.

10.4.3. Synthetic Dataset Generation

Since no publicly available benchmark currently exists for evaluating game-theoretic, risk-aware access-control systems based on temporal logic, a synthetic dataset was generated. The generated workload models realistic enterprise access requests involving different users, actions, protected objects, and execution contexts.

The workload consists of three categories of requests:

- Normal requests representing legitimate employee activities performed from trusted environments.
- Suspicious requests representing unusual but potentially legitimate behavior, such as repeated access from non-standard locations.
- Malicious requests representing insider attacks, including unauthorized deletion, repeated access to sensitive resources, and abnormal operations performed from external contexts.

To evaluate the adaptive behavior of RAFIA, repeated request sequences were generated so that the history of accepted and denied actions (τ^+ and τ^-) influenced subsequent authorization decisions.

The conditional probabilities $P(a|H)$ and $P(a|M)$ are estimated from the historical traces maintained by the prototype. Specifically, the frequencies of previously accepted and denied actions are used to approximate the likelihood that a given action is generated by an honest or a malicious user. These estimates are updated as additional requests are processed, enabling the Bayesian trust model to adapt over time.

10.4.4. Baseline Method

RAFIA was compared against a static-threshold access-control strategy. The baseline performs conventional policy verification followed by a fixed risk-threshold comparison, without considering historical traces, Bayesian trust updates, repeated actions, or game-theoretic reasoning.

This comparison highlights the contribution of RAFIA's adaptive learning mechanism and dynamic risk assessment beyond traditional static decision strategies.

10.4.5. Evaluation Metrics

The benchmark evaluates RAFIA from two complementary perspectives. From a security perspective, detection performance is measured using Accuracy, Precision, Recall, and F1-score. From an implementation perspective, computational efficiency is evaluated using average decision latency, throughput (requests processed per second), and peak memory consumption. Together, these metrics assess both the effectiveness of the proposed decision model and its practical suitability for deployment in real-time access-control environments.

10.4.6. Experimental Results

The benchmark was executed using workloads containing 100, 1,000, 5,000, and 10,000 access requests. Each workload was executed ten independent times using different random seeds. The reported results correspond to the mean values obtained across the ten executions.

Table 12 summarizes the computational performance of the RAFIA prototype under increasing workloads. As expected, the average latency increases gradually as more historical traces are processed, while throughput decreases with the workload size. Nevertheless, even for 10,000 requests the average authorization latency remains below 4 ms and the peak memory consumption remains below 10 MB, demonstrating that the proposed approach is suitable for real-time access-control applications.

Table 12. Computational performance of the RAFIA prototype.

Requests	Latency (ms)	Throughput (req/s)	Memory (MB)
100	0.55	1712	0.10
1000	0.80	1211	0.82
5000	1.75	563	4.12
10000	3.34	300	8.29

Figs. 13–15 illustrate the computational scalability of RAFIA. The latency grows approximately linearly with the number of processed requests due to the increasing size of the historical traces used during risk estimation. Conversely, throughput decreases as additional computations are performed for each authorization decision. Memory consumption also increases approximately linearly because the accepted and denied traces are maintained for adaptive learning. Despite

these increases, the prototype remains computationally lightweight, requiring less than 10 MB of memory while maintaining an average authorization latency below 4 ms decision latency for 10,000 requests.

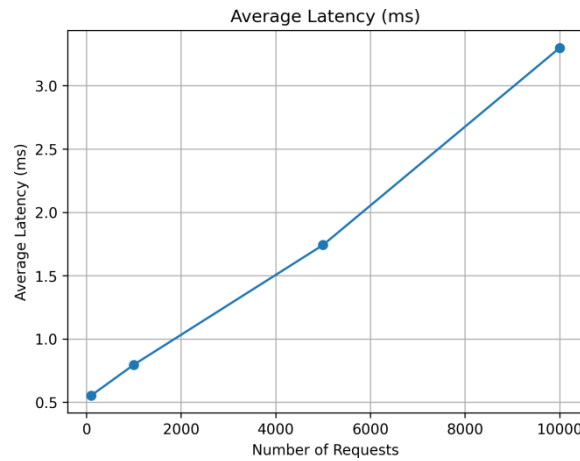


Fig. 13. Average authorization latency versus workload size.

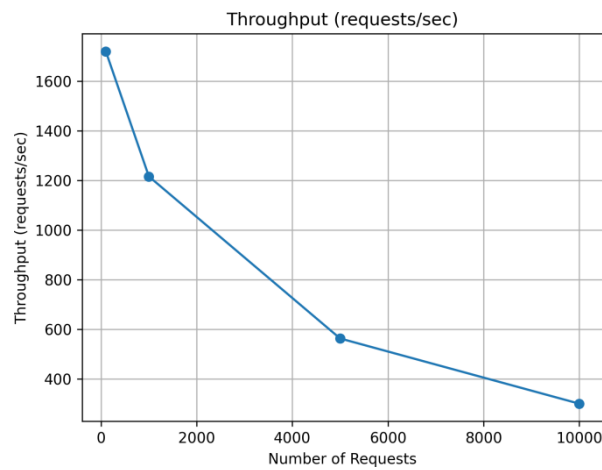


Fig. 14. System throughput under increasing workloads.

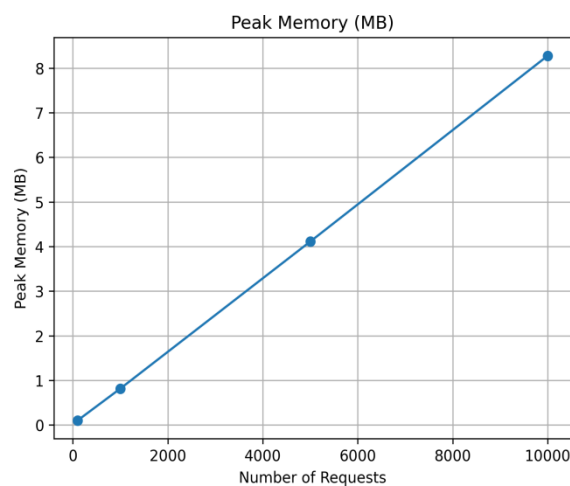


Fig. 15. Peak memory consumption of the RAFIA prototype

Table 13 and Figs. 16–17 compare RAFIA with a conventional static-threshold access-control strategy. RAFIA consistently achieves higher overall accuracy and substantially higher F1-score than the baseline. Although the baseline obtains higher precision by denying fewer requests, its recall is significantly lower, indicating that many malicious requests remain undetected.

Table 13. Detection performance comparison between RAFIA and the static-threshold baseline.

Metric	RAFIA	Static Baseline
Accuracy	0.615	0.587
Precision	0.614	0.783
Recall	1.000	0.453
F1-score	0.761	0.574

In contrast, RAFIA combines policy verification with historical traces, Bayesian trust updates, and game-theoretic decision making. This enables the framework to identify nearly all malicious requests while maintaining competitive overall accuracy. Consequently, RAFIA provides a better balance between security and usability than a conventional static-threshold approach.

10.4.7. Statistical Analysis

All benchmark results were obtained from ten independent executions for each workload size. The benchmark reports both the mean and the standard deviation of every performance metric, including latency, throughput, memory consumption, accuracy, precision, recall, and F1-score. The relatively small standard deviations observed across repeated executions indicate that the RAFIA prototype exhibits stable and reproducible behavior under identical experimental conditions.

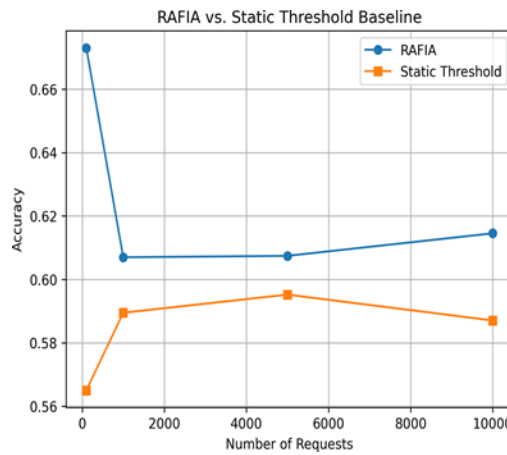


Fig. 16. Accuracy comparison between RAFIA and the static-threshold baseline.

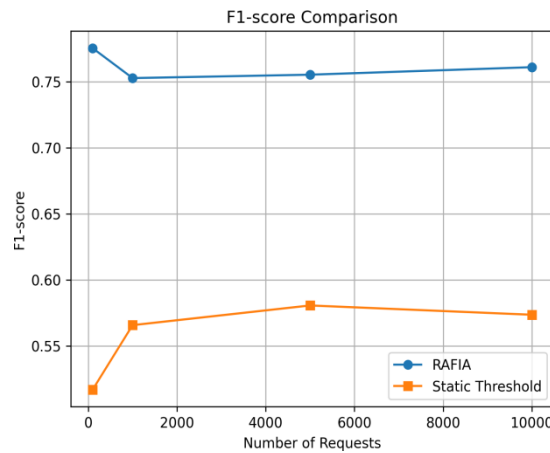


Fig. 17. F1-score comparison between RAFIA and the static-threshold baseline.

10.4.8. Interpretation of Results

The experimental evaluation demonstrates that incorporating game-theoretic reasoning and Bayesian trust updates improves the effectiveness of access-control decisions compared with a conventional static-threshold strategy. RAFIA consistently achieves higher recall and F1-score because authorization decisions consider not only the current request but also the historical traces and the estimated maliciousness of the user.

The computational results further indicate that these improvements are obtained with a relatively small computational overhead. Even for workloads containing 10,000 access requests, the average decision latency remains below 4 ms and the peak memory consumption remains below 10 MB. These results suggest that RAFIA is suitable for deployment in practical access-control systems requiring real-time authorization decisions.

Although the experiments rely on synthetically generated access traces due to the absence of publicly available datasets for risk-aware temporal access control, the benchmark models representative enterprise scenarios involving normal, suspicious, and malicious user behaviors. Future work will investigate the evaluation of RAFIA using real organizational access logs and publicly available insider-threat datasets whenever suitable annotations become available.

10.5. Practical Deployability

Table 14 illustrates the practical applicability of RAFIA across several application domains. The framework delivers enterprise-grade adaptive security without requiring significant infrastructure modifications, making it suitable for deployment in financial, healthcare, and critical infrastructure environments where both regulatory compliance and operational flexibility are essential.

- **Financial:** Prevents unusual account access, such as transfers exceeding \$50k from new devices. This helps detect and block potential fraud or unauthorized activities by monitoring atypical behavior patterns.
- **Healthcare:** Adapts to emergency access scenarios while ensuring compliance with HIPAA (Health Insurance Portability and Accountability Act) regulations. It allows quick access in urgent cases but prevents privacy violations by continuously monitoring access behaviors.
- **Startups:** Detects anomalous access to code repositories, such as mass downloads occurring prior to employee resignation. This helps identify potential data theft or leakage risks in early stages.

Table 14. RAFIA’s real-world applicability.

Domain	Solution
Financial	Blocks unusual account access (e.g., transfers over \$50k from new devices).
Healthcare	Adapts to emergency access patterns while preventing HIPAA violations.
Startups	Detects anomalous code repository access (e.g., mass downloads before employee resignation).

11. Discussion

The proposed RAFIA framework demonstrates that integrating formal policy specification, quantitative risk assessment, and game-theoretic reasoning provides a practical approach to mitigating insider threats in dynamic access-control environments. Unlike traditional static authorization mechanisms, RAFIA continuously adapts authorization decisions by combining Risk-LTL policy evaluation, historical execution traces, contextual information, Bayesian trust updates, and utility-based game-theoretic reasoning. This integration enables the framework to account for both the immediate characteristics of an access request and the accumulated behavioral history of the requesting user.

The experimental evaluation confirms the practical feasibility of the proposed approach. The benchmarking results show that RAFIA maintains low computational overhead while improving detection performance compared with a conventional static-threshold access-control strategy. In particular, the prototype achieved an average authorization latency of approximately 3.3 ms for workloads containing 10,000 access requests while maintaining low memory consumption. The comparison with the baseline further demonstrates that incorporating historical traces and adaptive risk evaluation substantially improves the F1-score and recall of malicious-request detection.

An important characteristic of RAFIA is its flexibility. The framework does not rely on a single predefined risk-estimation algorithm; instead, it separates the risk-computation mechanism from the decision-making process. Organizations may therefore employ application-specific risk models while preserving the same Risk-LTL policy language and game-theoretic authorization framework. This design enables RAFIA to be adapted to different operational environments without modifying its underlying architecture.

Recent industry reports indicate that insider threats continue to increase in both frequency and complexity, driven by the widespread adoption of cloud services, remote work, and generative AI. These trends reinforce the need for adaptive access-control mechanisms capable of continuously evaluating user behavior and contextual risk rather than relying on static authorization policies. Consequently, RAFIA provides a promising framework for addressing emerging insider-threat scenarios while maintaining flexible and risk-aware authorization decisions.

Despite these encouraging results, several limitations remain. The current evaluation relies on synthetically generated access traces because publicly available datasets combining temporal policies, contextual information, and risk annotations for insider-threat access control remain scarce. Moreover, the present framework models interactions between a single organization and a single user and therefore does not explicitly address coordinated or collaborative insider attacks. Future work will investigate the use of real organizational audit logs, more sophisticated probability-estimation techniques, and extensions supporting multi-user collaborative threat models.

12. Conclusion

Insider attacks are subtle and difficult to detect, and they are harmful to organizations. Many promising techniques have been proposed to mitigate them; however, the problem remains unresolved. Any contribution in this area is highly encouraged. This paper proposed RAFIA, a game-theoretic risk-based framework for detecting and preventing high-risk insider attacks. To use RAFIA, security managers need to specify their security policies using a linear temporal logic called Risk-LTL. They also specify a risk evaluation function that will be used to evaluate the risk associated with incoming access requests. Moreover, we design the access-control model as a multi-stage game between the organization and users, with the objective of supporting risk-aware detection and management of potentially malicious behavior. A prototype implementation of RAFIA was developed to demonstrate the feasibility of the proposed framework. Experimental evaluation using synthetic workloads containing up to 10,000 access requests demonstrated that RAFIA can efficiently support risk-aware authorization decisions while maintaining low computational overhead. The results also demonstrate improved detection performance compared with a conventional static-threshold access-control strategy, confirming the practical applicability of the proposed framework. One limitation of the current RAFIA framework is that access-control decisions are primarily modeled from the perspective of an individual subject interacting with the system. Consequently, collaborative insider attacks involving multiple coordinated users are not explicitly represented in the current formulation. In addition, the current evaluation relies on synthetically generated access traces because publicly available datasets specifically designed for risk-aware temporal access control remain limited. As future work, we plan to extend RAFIA to support multi-agent threat models and collaborative insider attack detection, investigate the use of real organizational audit logs, and explore more advanced probability-estimation techniques, thereby enhancing its applicability to more complex organizational environments.

All the Declarations and Statements

Author Contributions Statement

Hala Yousif Mohamed Ahmed – Methodology, initial RAFIA framework development, software implementation, experimental evaluation, literature review, and writing – original draft.

Mohamed Mejri – Conceptualization, supervision, RAFIA framework refinement, Risk-LTL formalization, mathematical modeling, and writing – review & editing.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This research received no external funding.

Data Availability Statement

Data sharing is not applicable.

Ethical Declarations

This study does not involve human participants or animals.

Acknowledgments

The authors sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.

Declaration of Generative AI in Scholarly Writing

During the preparation of this manuscript, generative AI tools were used solely to improve the language and readability of the manuscript. The authors carefully reviewed and edited the AI-assisted content and accept full responsibility for the accuracy, originality, and integrity of the final manuscript.

Abbreviations

The following abbreviations are used in this manuscript:

ABAC – Attribute-Based Access Control

ACD – Access Control Domain

AI – Artificial Intelligence
 CIA – Confidentiality, Integrity, and Availability
 CSV – Comma-Separated Values
 DBMS – Database Management System
 DLBAC – Deep Learning Based Access Control
 GDPR – General Data Protection Regulation
 GUI – Graphical User Interface
 HIPAA – Health Insurance Portability and Accountability Act
 IoT – Internet of Things
 IP – Internet Protocol
 LLM – Large Language Model
 LTL – Linear Temporal Logic
 NE – Nash Equilibrium
 RAFIA – Risk-Based Framework for Insider Threat Mitigation
 RBAC – Role-Based Access Control
 Risk-LTL – Risk-Aware Linear Temporal Logic
 RM – Risk Model

Appendix A/B/C..., with appendix title

N/A

References

- [1] M. Stamp, *Information Security: Principles and Practice*, 3rd ed. John Wiley & Sons, 2021.
- [2] Verizon, “Data Breach Investigations Report.” 2024. Accessed: Jun. 03, 2025. [Online]. Available: <https://www.verizon.com/business/resources/reports/dbir/>.
- [3] “2024 Insider Threat Report.” Accessed: Jun. 20, 2025. [Online]. Available: <https://www.cybersecurity-insiders.com/2024-insider-threat-report/>
- [4] Exabeam, “2025 Insider Threat Report: From Humans to AI Agents.” 2025. Accessed: Jun. 20, 2025. [Online]. Available: <https://www.exabeam.com/>
- [5] Ponemon Institute, “Cost Of Insider Risks Global Report,” Ponemon Institute, 2023. Accessed: Jun. 03, 2025. [Online]. Available: <https://ponemonsullivanreport.com/2023/10/cost-of-insider-risks-global-report-2023>.
- [6] Gartner, “Cybersecurity Insights & Solutions for CISOs & Leaders.” 2023. Accessed: Jul. 01, 2026. [Online]. Available: <https://www.gartner.com/en/cybersecurity>
- [7] National Institute of Standards and Technology (Corporate Author), “Security and Privacy Controls for Information Systems and Organizations,” National Institute of Standards and Technology, Gaithersburg, MD, USA, Special Publication 800-53 Revision 5, 2020. doi: 10.6028/NIST.SP.800-53r5.
- [8] V. Suhendra, “A Survey on Access Control Deployment,” in *Communications in Computer and Information Science*, vol. 259, T. Kim, H. Adeli, W. Fang, J. G. Villalba, K. P. Arnett, and M. K. Khan, Eds., Berlin/Heidelberg, Germany: Springer, 2011, pp. 11–20.
- [9] P.-C. Cheng, et al., “Fuzzy multi-level security: An experiment on quantified risk-adaptive access control,” in *2007 IEEE Symposium on Security and Privacy (SP’07)*, IEEE, 2007, pp. 222–230. doi: 10.1109/SP.2007.21.
- [10] R. A. Shaikh, K. Adi, and L. Logrippo, “Dynamic risk-based decision methods for access control systems,” *Comput. Secur.*, vol. 31, no. 4, pp. 447–464, 2012, doi: 10.1016/j.cose.2012.02.006.
- [11] Díaz-López, Daniel, Dólera-Tormo, Ginés, Félix Gómez-Mármol, and Gregorio Martínez-Pérez, “Dynamic Counter-Measures for Risk-Based Access Control Systems: An Evolutionary Approach,” *Future Gener. Comput. Syst.*, vol. 55, pp. 321–335, 2016, doi: <https://doi.org/10.1016/j.future.2014.10.012>.
- [12] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” *IEEE Commun. Surv. Tutor.*, vol. 16, no. 1, pp. 414–454, 2013, doi: 10.1109/SURV.2013.042313.00197.
- [13] H. Ding, C. Peng, Y. Tian, and S. Xiang, “A Game Theoretical Analysis of Risk Adaptive Access Control for Privacy Preserving,” in *2019 International Conference on Networking and Network Applications (NaNA)*, IEEE, 2019, pp. 253–258. doi: 10.1109/NaNA.2019.00052.
- [14] Lei Zhang; A. Brodsky; S. Jajodia, “Toward information sharing: benefit and risk access control (BARAC),” in *Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY’06)*, IEEE Computer Society, 2006, pp. 9–17. doi: 10.1109/POLICY.2006.36.
- [15] L. Pokorádi, “Application of fuzzy set theory for risk assessment,” *J. KONBiN*, vol. 14, no. 1, pp. 187–196, 2010.
- [16] H. F. Atlam, M. A. Azad, M. O. Alassafi, A. A. Alshdadi, and A. Alenezi, “Risk-Based Access Control Model: A Systematic Literature Review,” *Future Internet*, vol. 12, no. 6, p. 103, 2020, doi: 10.3390/fi12060103.
- [17] D. Akinwumi, G. Iwasokun, B. Alese, and S. Oluwadare, “A review of game theory approach to cyber security risk management,” *Niger. J. Technol.*, vol. 36, no. 4, pp. 1271–1285, 2017, doi: 10.4314/njt.364.1502.
- [18] Y. Wang, L. Tian, and Z. Chen, “Game Analysis of Access Control Based on User Behavior Trust,” *Information*, vol. 10, no. 4, p. 132, 2019, doi: 10.3390/info10040132.

- [19] Office of the Privacy Commissioner of Canada, "Combination of weaknesses led to massive data breach at Desjardins." Accessed: Jun. 23, 2026. [Online]. Available: https://www.priv.gc.ca/en/opc-news/news-and-announcements/2020/nr-c_201214/.
- [20] F. L. Greitzer, L. J. Kangas, C. F. Noonan, R. E. Hohimer, and A. C. Dalton, "Identifying At-Risk Employees: Modeling Psychosocial Precursors of Potential Insider Threats," in *2012 45th Hawaii International Conference on System Sciences*, Maui, HI, USA: IEEE, Jan. 2012, pp. 2392–2401. doi: 10.1109/HICSS.2012.309.
- [21] S. Tadelis, *Game Theory: An Introduction*. Princeton, NJ: Princeton University Press, 2013.
- [22] E. Ho, A. Rajagopalan, A. Skvortsov, S. Arulampalam, and M. Piraveenan, "Game Theory in Defence Applications: A Review," *Sensors*, vol. 22, no. 3, p. 1032, 2022, doi: 10.3390/s22031032.
- [23] A. K. Malik *et al.*, "From Conventional to State-of-the-Art IoT Access Control Models," *Electronics*, vol. 9, no. 10, p. 1693, 2020, doi: 10.3390/electronics9101693.
- [24] K. Ma, G. Yang, and Y. Xiang, "RCBAC: A risk-aware content-based access control model for large-scale text data," *J. Netw. Comput. Appl.*, vol. 167, p. 102733, 2020, doi: 10.1016/j.jnca.2020.102733.
- [25] D. R. Dos Santos, C. M. Westphall, and C. B. Westphall, "A dynamic risk-based access control architecture for cloud computing," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, IEEE, 2014, pp. 1–9. doi: 10.1109/NOMS.2014.6838319.
- [26] B. M. Babu and M. S. Bhanu, "Prevention of insider attacks by integrating behavior analysis with risk based access control model to protect cloud," *Procedia Comput. Sci.*, vol. 54, pp. 157–166, 2015, doi: 10.1016/j.procs.2015.06.018.
- [27] Y. Li, H. Sun, Z. Chen, J. Ren, and H. Luo, "Using trust and risk in access control for grid environment," in *2008 International Conference on Security Technology*, IEEE, 2008, pp. 13–16. doi: 10.1109/SecTech.2008.50.
- [28] R. McGraw, "Risk-adaptable access control (radac)," in *Privilege (Access) Management Workshop. NIST–National Institute of Standards and Technology–Information Technology Laboratory*, 2009, pp. 55–58.
- [29] R. A. Shaikh, K. Adi, L. Logrippo, and S. Mankovski, "Risk-based decision method for access control systems," in *2011 Ninth Annual International Conference on Privacy, Security and Trust*, IEEE, 2011, pp. 189–192. doi: 10.1109/PST.2011.5971982.
- [30] L. Rajbhandari and E. A. Snekenes, "Using game theory to analyze risk to privacy: An initial insight," in *IFIP PrimeLife International Summer School on Privacy and Identity Management for Life*, Springer, 2010, pp. 41–51. doi: 10.1007/978-3-642-20769-3_4.
- [31] N. N. Diep, *et al.*, "Enforcing access control using risk assessment," in *Fourth European Conference on Universal Multiservice Networks (ECUMN'07)*, IEEE, 2007, pp. 419–424. doi: 10.1109/ECUMN.2007.19.
- [32] N. Metoui, M. Bezzi, and A. Armando, "Trust and risk-based access control for privacy preserving threat detection systems," in *International Conference on Future Data and Security Engineering*, Springer, 2016, pp. 285–304. doi: 10.1007/978-3-319-48057-2_20.
- [33] H. Khambhammettu, S. Boulares, K. Adi, and L. Logrippo, "A framework for risk assessment in access control systems," *Comput. Secur.*, vol. 39, pp. 86–103, 2013, doi: 10.1016/j.cose.2013.03.010.
- [34] M. Sharma, Y. Bai, S. Chung, and L. Dai, "Using risk in access control for cloud-assisted ehealth," in *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, IEEE, 2012, pp. 1047–1052. doi: 10.1109/HPCC.2012.153.
- [35] H. F. Atlam, R. J. Walters, G. B. Wills, and J. Daniel, "Fuzzy logic with expert judgment to implement an adaptive risk-based access control model for IoT," *Mob. Netw. Appl.*, pp. 1–13, 2019, doi: 10.1007/s11036-019-01214-w.
- [36] F. K. Dankar and R. Badji, "A risk-based framework for biomedical data sharing," *J. Biomed. Inform.*, vol. 66, pp. 231–240, 2017, doi: 10.1016/j.jbi.2017.01.012.
- [37] M. Abomhara, G. M. Køien, V. A. Oleshchuk, and M. Hamid, "Towards Risk-aware Access Control Framework for Healthcare Information Sharing," in *ICISSP*, 2018, pp. 312–321. doi: 10.5220/0006608103120321.
- [38] C. A. B. De Carvalho, R. M. de Castro Andrade, N. Agoulmine, and M. F. de Castro, "Detection of Access Control Violations in the Secure Sharing of Cloud Storage.," in *CLOSER*, 2018, pp. 124–135. doi: 10.5220/0006698701240135.
- [39] P. A. Evina, F. L. Ayachi, F. Jaidi, and A. Bouhoula, "Anomalies Correlation for Risk-Aware Access Control Enhancement.," in *ENASE*, 2018, pp. 299–304. doi: 10.5220/0006766802990304.
- [40] X. Chen, "Risk-based Access Control Model for Hospital Information Systems," *Front. Comput. Intell. Syst.*, 2023, doi: 10.54097/fcis.v2i3.5315.
- [41] M. N. Nobl, R. Krishnan, Y. Huang, M. Shakarami, and R. Sandhu, "Toward Deep Learning Based Access Control," in *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, in CODASPY '22. ACM, 2022, pp. 143–154. doi: 10.1145/3508398.3511497.
- [42] J. Yin, *et al.*, "A Heterogeneous Graph-Based Semi-Supervised Learning Framework for Access Control Decision-Making," *World Wide Web*, vol. 27, no. 4, p. 35, 2024, doi: 10.1007/s11280-024-01275-2.
- [43] S. Yarram, N. Dasari, S. B. Seshagani, and P. Ganguly, "Privacy-Preserving Healthcare Data Security Using Large Language Models and Adaptive Access Control," in *2025 IEEE World AI IoT Congress (AIIoT)*, 2025. doi: 10.1109/AIIoT65859.2025.11105296.

Authors' Profiles



Hala Yousif Mohamed Ahmed is currently pursuing a Ph.D. in Computer Science. She received her B.Sc. (Honours) and M.Sc. degrees in Computer Science from the University of Khartoum, Sudan. Her research interests include cybersecurity, Internet of Things (IoT) security, AI-driven threat detection, game theory, and risk-aware access control.



Mohamed Mejri received the Engineering degree in Computer Science from the National School of Computer Science (ENSI), Tunisia, the M.Sc. degree in Computer Science from Laval University, Canada, and the Ph.D. degree in the specification and analysis of cryptographic protocols from the Department of Computer Science and Software Engineering, Laval University. He is currently a Full Professor and the Director of the Laboratory for Computer Security in the Department of Computer Science and Software Engineering, Laval University. His research interests include computer security, formal methods, and software engineering.

How to cite this paper

Hala Yousif Mohamed Ahmed, Mohamed Mejri, "RAFIA: A Game-theoretic Risk-based Framework for Insider Threat Mitigation", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.4, pp. 197-232, 2026. DOI:10.5815/ijwmt.2026.04.12