The ISC Int'l Journal of Information Security
ISeCure

July 2026, Volume 18, Number 2 (pp. 69–83)
http://www.isecure-journal.org

GMASO: A Graph-based Multi-Agent Security Optimizer for Threat-Specific Countermeasure Selection in Mission-Critical Systems

Sajed Yousefi Mashhour [1], Motahareh Dehghan [2,*], Babak Sadeghian [3], Alireza Hashemi Golpayegani [3]

[1] *Department of Management, Science and Technology, Amirkabir University of Technology, Tehran, Iran*
[2] *Department of Industrial Engineering, Tarbiat Modares University, Tehran, Iran*
[3] *Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran*

ARTICLE INFO.

Article history:
Received:
Revised:
Accepted:
Published Online:

Keywords:
Cybersecurity Optimization, Multi-Agent Modeling, Dynamic Risk Assessment, Countermeasure Prioritization, Mission-Critical Systems

Type:

doi:

Abstract

Existing cybersecurity frameworks suffer from static threat assessment, inadequate modeling of system dependencies, oversimplified risk propagation, and inflexible countermeasure selection. This study presents a data-driven, multi-agent decision-support framework that optimizes countermeasure selection under operational constraints. The approach employs a dynamic graph structure representing relationships among missions, tasks, assets, threats, vulnerabilities, and countermeasures, with weighted dependencies across confidentiality, integrity, and availability dimensions. The framework comprises a modular architecture of ten specialized agents—namely the Main (serving as the Environment Controller agent), Node, Edge, Mission, Task, Asset, Threat, Vulnerability, Countermeasure, and Mapping Agents—organized into four functional categories (Control, Structural, Mission-Centric, and Security-Focused). These agents collaboratively operate through four sequential phases: (1) Data Gathering and Graph Construction, (2) Weight Propagation and Risk Assessment, (3) Multi-Criteria Optimization, and (4) Implementation and Reassessment. Experimental results demonstrate substantial improvements in risk mitigation efficiency and resource allocation compared to conventional approaches, enabling organizations to dynamically align security investments with evolving threats, mission priorities, and budget constraints while maintaining operational continuity.

1 Introduction

The rapid expansion and interconnectivity of digital infrastructures have enabled unprecedented operational capabilities but also exposed organizations to increasingly sophisticated, multi-vector cyber threats. Static defenses, uniform control assessments, and inadequate modeling of complex dependencies leave critical gaps in protection. Vulnerabilities in a single asset can propagate across interconnected missions, tasks, and resources, threatening

* Corresponding author.
Email addresses: sajed.yousefi@aut.ac.ir, m_dehghan@modares.ac.ir, basadegh@aut.ac.ir, sa.hashemi@aut.ac.ir
ISSN: 2008-2045 © 2026 ISC. All rights reserved.

continuity and increasing systemic risk. In mission-critical environments—such as defense operations, critical infrastructure, and enterprise systems—such propagation can compromise not only individual components but entire operational objectives, making context-agnostic security measures insufficient.

Modern cyber-physical systems integrate IT and OT infrastructures, creating complex interdependencies where vulnerabilities can propagate across missions, tasks, and resources, threatening continuity and increasing systemic risk, making context-agnostic security measures insufficient. Our previous work [1] addressed this challenge by extending RiskMAP with a mission-centric countermeasure selection framework that evaluates security measures based on mission alignment and operational impact.

Existing cybersecurity approaches often fail to adapt to evolving threats, to model system components contextually, and to apply control metrics that account for mission relevance, operational cost, and contextual effectiveness. Although advances in multi-agent systems and graph-based risk modeling have improved detection and visualization, they often remain reactive and lack integrated, mission-aware countermeasure optimization. Specifically, while existing multi-agent systems have enhanced distributed detection, they frequently lack the systemic perspective needed for mission alignment. Similarly, while graph-based models capture dependencies, they often fail to offer multi-criteria decision support for countermeasure selection under operational and budget constraints. This gap necessitates a framework that combines structured dependency modeling with collaborative, mission-centric optimization.

To address these limitations, we propose the Graph-based Multi-Agent Security Optimizer (GMASO), a decision-support framework combining structured dependency modeling with hierarchical, specialized agents. The architecture employs a directed, CIA-weighted graph—where CIA denotes the classical Confidentiality, Integrity, and Availability dimensions of cybersecurity—to capture fine-grained relationships among missions, tasks, assets, threats, vulnerabilities, and countermeasures. These CIA weights quantify the relative impact of each connection along these three security dimensions, guiding agent-level reasoning in prioritizing and selecting optimal countermeasures. Ten agent types operate collaboratively for risk assessment, prioritization, and dynamic countermeasure selection under operational and budget constraints. This design directly addresses the identified gap: the CIA-weighted dependency graph provides the systemic perspective needed to model propagation across missions, while the ten specialized agents enable modular, collaborative risk assessment and automated multi-criteria optimization that existing approaches lack.

GMASO integrates real-time adaptability, mission-centric prioritization, multi-criteria optimization, and bidirectional impact analysis, enabling precise mapping between threats and countermeasures. This improves resource efficiency and operational alignment compared to conventional methods. The framework, including simulation tools and datasets, is publicly available for reproducibility and extension.

The key contributions of this work are as follows:

- **Mission-aware countermeasure selection:** GMASO explicitly incorporates mission objectives and task criticality into the countermeasure selection process, directly addressing the inadequacy of static, context-agnostic defenses. This ensures that security investments and responses are aligned with operational importance.
- **CIA-weighted dependency modeling:** A directed graph that quantifies inter-entity relationships using Confidentiality, Integrity, and Availability weights, enabling granular assessment of propagation and criticality across interconnected missions, tasks, and assets—addressing the gap in systemic vulnerability modeling.
- **Hierarchical multi-agent architecture:** A ten-agent structure that collaboratively performs contextual risk evaluation, weight normalization, and adaptive countermeasure optimization under operational and budget constraints, bridging the gap between reactive detection and proactive, mission-aligned decision support.
- **Bidirectional impact analysis:** A mechanism that models both forward (threat-to-mission) and backward (mission-to-asset) dependencies for accurate impact propagation and prioritization of security actions, overcoming the limitations of unidirectional risk assessment in prior work.

The remainder of this paper reviews related methodologies, details the GMASO system architecture, presents empirical validation in an enterprise scenario, and discusses implications and future work.

2 Related Work

The dynamic and interconnected nature of modern cyber systems has exposed significant limitations in existing cybersecurity frameworks, particularly in their ability to adapt to evolving threats and model complex system interdependencies. Static threat

assessments, oversimplified risk propagation, and generic countermeasure evaluations often result in inefficient resource allocation and persistent security gaps. Recent advancements in multi-agent systems (MAS), graph-based security modeling, probabilistic risk assessment, multi-objective optimization, game-theoretic modeling, and real-time risk assessment have attempted to address these challenges, but critical limitations remain.

Multi-agent systems, as explored by Viduto *et al.* [2], Kotenko and Doynikova [3], and Shameli-Sendi *et al.* [4], offer distributed risk assessment and adaptive control strategies. However, scalability challenges, coordination overhead, and limited mission-aware countermeasure selection hinder their practical deployment in large-scale environments. Graph-based security models, such as those by Nespoli *et al.* [5], Tariq *et al.* [6], and Roy *et al.* [7], provide valuable dependency visualization and attack propagation mapping. Yet, static representations and computational inefficiencies limit their ability to address dynamically shifting threats and evolving system configurations. A broader survey of attacker–defender game models by Hausken *et al.* [8] further highlights recurring gaps across these paradigms, including incomplete information structures, limited resilience modeling, and the absence of mission-aware countermeasure selection in both MAS and graph-based approaches.

Probabilistic frameworks, including Wang *et al.* [9], introduce quantitative metrics for security hardening but often fail to capture real-time threat dynamics or layered system dependencies. Multi-objective optimization techniques, such as those by Dewri *et al.* [10] and Granadillo *et al.* [11], balance risk reduction, cost, and service quality but struggle with scalability and nuanced threat-specific effectiveness in resource-constrained enterprise contexts. Game-theoretic approaches, such as the Stackelberg security game proposed by Ait Temghart *et al.* [12] for cloud environments, model bounded rationality and preference uncertainty through a modified quantal response formulation; however, their defender strategies remain confined to cloud-specific interaction models and do not generalize to mission-critical, graph-structured enterprise systems. Real-time adaptive risk assessment, as demonstrated by Shameli-Sendi *et al.* [13] and Sonmez and Kilic [14], enables dynamic response to emergent threats but is limited by computational bottlenecks and incomplete modeling of propagation effects. More recently, Goel *et al.* [15] applied reinforcement learning within a Stackelberg attacker–defender framework to optimize edge-blocking strategies in dynamic Active Directory graphs; while this advances scalable cyber defense, it does not address mission-centric dependency modeling or multi-criteria countermeasure selection under operational constraints.

Despite these advancements, existing methodologies lack an integrated approach that dynamically represents mission objectives, quantifies layered dependencies, incorporates a multi-agent approach, and robustly adapts to new threat vectors in real time. While our previous work [1] introduced mission-centric countermeasure selection using RiskMAP extension and multi-criteria evaluation, it did not address dynamic graph-based dependency modeling or hierarchical multi-agent coordination. The proposed GMASO framework addresses these gaps by unifying graph-based modeling, hierarchical multi-agent coordination, and dynamic optimization.

3 Methodology

This section presents the GMASO framework's systematic approach to mission-centric cybersecurity optimization, detailing the multi-agent architecture, weight propagation mechanisms, optimization criteria, and performance evaluation metrics.

3.1 Overview

GMASO combines risk assessment, weight propagation, and multi-criteria optimization across the confidentiality, integrity, and availability (CIA) dimensions.

System entities are modeled as nodes and their dependencies as CIA-weighted edges, quantifying criticality and propagation strength. This structure captures organizational security relations in detail while maintaining computational efficiency. Specialized autonomous agents collaboratively perform risk evaluation, normalize dependency weights, and optimize countermeasure selection to support mission-centric decision-making.

3.2 Multi-Agent Architecture and Operational Phases

This subsection describes the hierarchical agent structure and the four-phase operational workflow that enables dynamic risk assessment and countermeasure optimization.

GMASO employs ten specialized agents—namely the Main (Environment Controller), Node, Edge, Mission, Task, Asset, Threat, Vulnerability, Countermeasure, and Mapping Agents—grouped into four functional categories under the supervision of the Environment Controller Agent, which serves as the **Main coordinating agent type**, as illustrated in Figure 1. The four categories are:

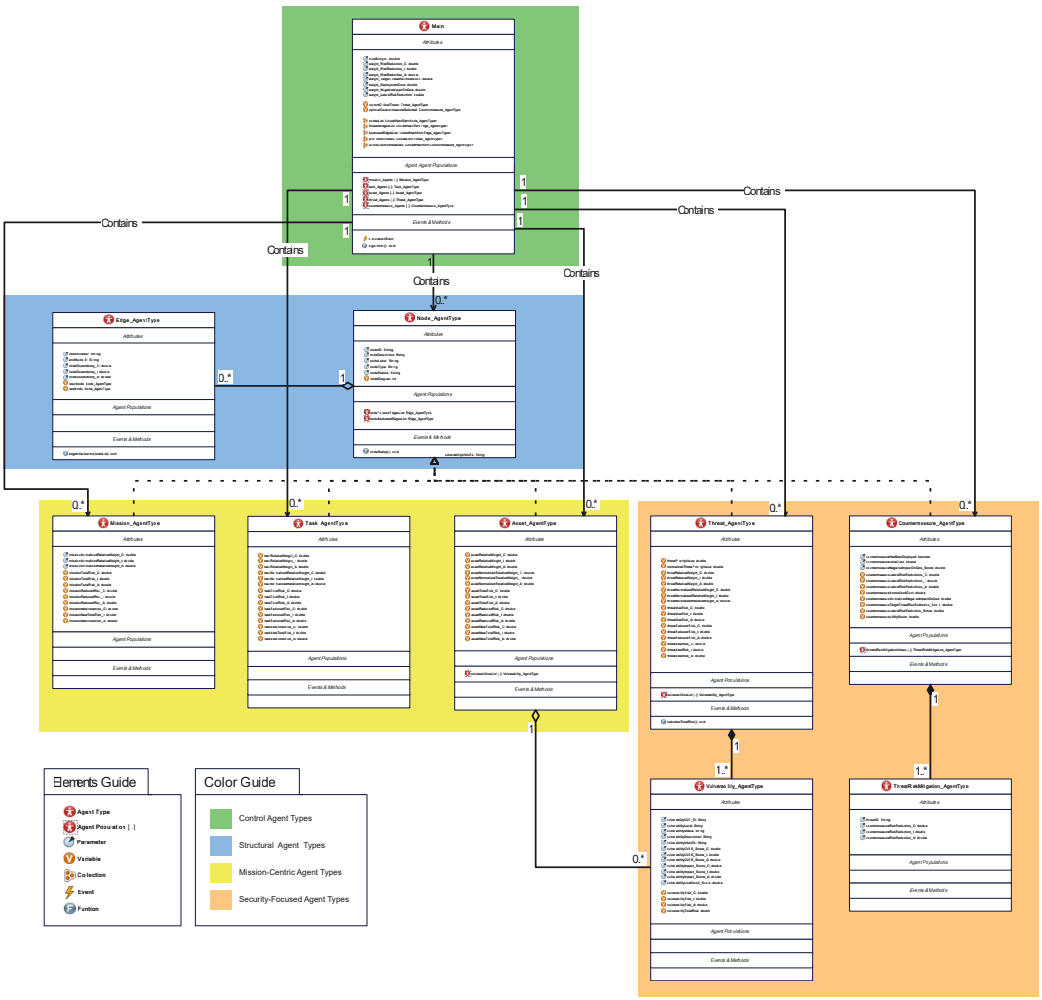

Figure 1. Graph-Based Multi-Agent System Architecture. The **Environment Controller** (top, highlighted) serves as the main supervisory agent coordinating ten specialized agents organized into four functional categories: **Control** (Environment Controller), **Structural** (Node, Edge), **Mission-Centric** (Mission, Task, Asset, Mapping), and **Security-Focused** (Threat, Vulnerability, Countermeasure). Arrows indicate hierarchical supervision and inter-agent communication flows.

- **Control**: Main Agent (supervisory coordination)
- **Structural**: Node and Edge Agents (CIA-weighted graph construction)
- **Mission-Centric**: Mission, Task, and Asset Agents (mission modeling and asset tracking)
- **Security-Focused**: Threat, Vulnerability, and Countermeasure Agents (risk assessment and mitigation)

The architecture operates through four phases reflecting the GMASO workflow as depicted in Figure 2:

Phase 1 – Data Gathering and Graph Construction: Agents initialize mission, task, asset, vulnerability, threat, and countermeasure entities, forming CIA-weighted dependencies that represent organizational structures and relationships.

Phase 2 – Weight Propagation and Risk Evaluation: CIA weights and mission importance values flow across hierarchy layers using RiskMAP relationships. Bidirectional propagation quantifies how vulnerabilities influence mission outcomes, while normalization ensures consistent scale and comparability.

Phase 3 – Multi-Criteria Optimization: The optimization framework evaluates each countermeasure cm_j through four complementary criteria that collectively address both security effectiveness and operational feasibility. Each criterion captures a distinct aspect of countermeasure value, enabling holistic decision-making under resource constraints:

Phase 4 – Implementation and Reassessment: Selected countermeasures are applied in simulation; residual risks are recalculated across graph

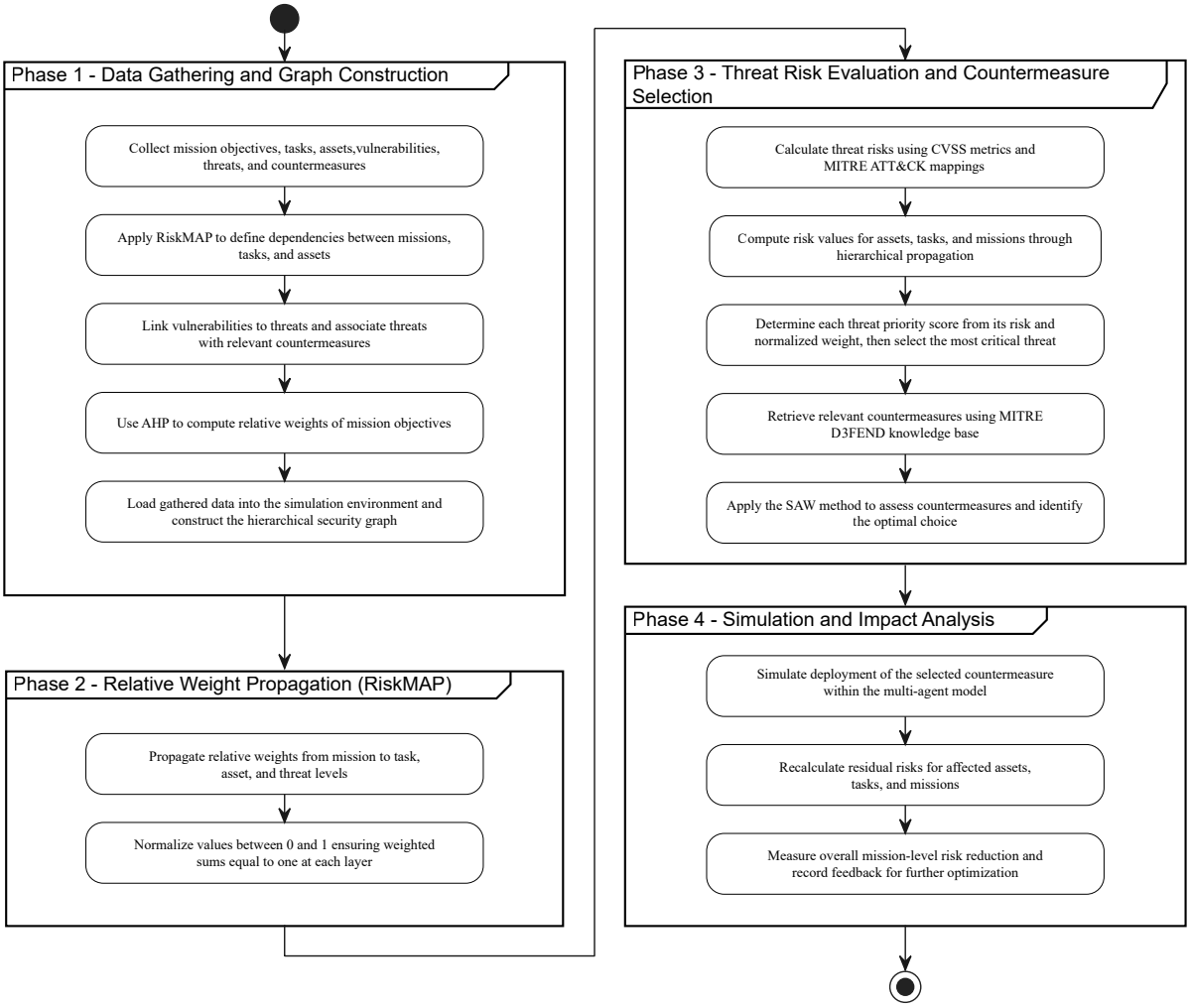

Figure 2. GMASO framework illustrating the four integrated phases. High-level architecture of the GMASO framework, illustrating the integration of four phases: graph construction, threat modeling, multi-criteria optimization, and countermeasure deployment. Detailed mathematical formulations for each phase are provided in Section 3.2.

layers, producing refined indicators for system performance and mission-level improvement.

- **Threat Risk Mitigation (TR):** Quantifies the direct reduction in threat exposure achieved by implementing the countermeasure.
- **Cost Efficiency (C):** Normalizes countermeasure costs to enable comparison across different price ranges, with higher values indicating more cost-effective solutions.
- **Quality-of-Service Impact (Q):** Assesses both direct and indirect operational impacts of countermeasure deployment on system performance.
- **Lateral Risk Reduction (LRR):** Measures how effectively a countermeasure prevents threat propagation across network boundaries.

The following subsections formalize each criterion mathematically.

3.2.0.1 Threat Risk Mitigation (TR).

This criterion quantifies the direct reduction in threat exposure achieved by implementing countermeasure cm_j. Let $TH(cm_j)$ denote the set of threats addressed by cm_j. The threat mitigation score aggregates the normalized threat weight, the inherent risk level of each threat, and the mitigation effectiveness of the countermeasure:

$$TR(cm_j) = \sum_{z \in TH(cm_j)} \sum_{d \in \{C,I,A\}} \hat{W}^{Th}_{z,d} \cdot R^{Th}_{z,d} \cdot M^{th}_{jd}, \quad (1)$$

where $\hat{W}^{Th}_{z,d}$ denotes the normalized weight of threat z in dimension d, $R^{Th}_{z,d}$ denotes the inherent risk level of threat z in dimension d, and M^{th}_{jd} denotes the mit-

igation effectiveness of countermeasure cm_j against threat z in dimension d.

3.2.0.2 Cost Efficiency (C).

This criterion normalizes countermeasure costs to enable comparison across different price ranges. Given the cost $C(cm_j)$ of countermeasure cm_j [16], the normalized cost efficiency is:

$$\hat{C}(cm_j) = 1 - \frac{C(cm_j)}{\max_{j'} C(cm_{j'})}. \quad (2)$$

Higher values of $\hat{C}(cm_j)$ indicate more cost-effective solutions.

3.2.0.3 Quality-of-Service Impact (Q).

This criterion assesses both direct and indirect operational impacts of countermeasure deployment [16]. Direct service quality (DSQ) measures immediate performance degradation on affected systems, while indirect service quality (ISQ) captures cascading effects on dependent systems. The overall QoS impact is computed as:

$$Q(cm_j) = \alpha \cdot DSQ(cm_j) + (1-\alpha) \cdot ISQ(cm_j), \quad (3)$$

where $\alpha \in [0,1]$ balances direct versus indirect service quality considerations, $DSQ(cm_j)$ denotes the direct service quality impact of cm_j, and $ISQ(cm_j)$ denotes its indirect service quality impact.

3.2.0.4 Lateral Risk Reduction (LRR).

This criterion quantifies how effectively a countermeasure prevents threat propagation across network boundaries. Let $N(z)$ denote the set of threats reachable from threat z through lateral movement paths. The lateral risk reduction $LR_c(cm_j)$ is calculated as the relative reduction in risk across reachable threats, where $R^{Th,post}_{z',d}$ represents the post-deployment risk level of threat z':

$$LR_c(cm_j) = \frac{\sum_{z \in TH(cm_j)} \sum_{z' \in N(z)} \left(R^{Th}_{z,d} - R^{Th,post}_{z',d} \right)}{\sum_{z \in TH(cm_j)} \sum_{z' \in N(z)} R^{Th}_{z,d}}, \quad (4)$$

which measures the proportional reduction in lateral threat diffusion potential.

3.3 Hierarchical Weight Propagation

This subsection establishes the mathematical foundation for quantifying mission-to-threat dependencies through normalized weight propagation across organizational layers.

Before detailing the weight propagation mechanism, we briefly introduce the foundational methodologies employed in our framework:

RiskMAP [17] is a mission-aware risk assessment methodology that aligns security priorities with organizational mission objectives. It constructs dependency graphs linking missions to tasks, assets, and threats, enabling risk calculations that reflect mission criticality rather than treating all assets uniformly. Our previous work [1] extended RiskMAP with a mission-centric countermeasure selection framework that evaluates security controls using multi-criteria decision-making based on mission alignment, CIA triad impact, and operational constraints.

Analytic Hierarchy Process (AHP) [17] is a multi-criteria decision-making technique that decomposes complex decisions into pairwise comparisons. In our context, AHP is used to elicit expert judgments on the relative importance of CIA attributes for each mission.

Consistency Ratio (CR) [17] measures the logical consistency of pairwise comparison matrices in AHP. A CR value below 0.1 indicates acceptable consistency, ensuring that expert judgments are coherent and reliable. Matrices with $CR \geq 0.1$ are flagged for re-evaluation to maintain the integrity of the weight elicitation process.

Our framework integrates these methodologies as follows: RiskMAP provides the architectural foundation for mission-aware risk modeling, AHP elicits mission-specific CIA weights from domain experts, and CR validation ensures the reliability of these weights before propagation through the dependency graph.

The dependency graph is organized into four hierarchical layers, each containing a finite set of entities:

- **Mission layer:** $\mathcal{M} = \{j \in \mathbb{N} \mid 1 \leq j \leq N_{\text{MO}}\}$, where N_{MO} is the total number of mission objectives.
- **Task layer:** $\mathcal{T} = \{i \in \mathbb{N} \mid 1 \leq i \leq N_T\}$, where N_T is the total number of tasks.
- **Asset layer:** $\mathcal{A} = \{k \in \mathbb{N} \mid 1 \leq k \leq N_A\}$, where N_A is the total number of assets.
- **Threat layer:** $\mathcal{TH} = \{z \in \mathbb{N} \mid 1 \leq z \leq N_{Th}\}$, where N_{Th} is the total number of threats.

We denote the set of all layers as $\mathcal{L} \in \{\mathcal{M}, \mathcal{T}, \mathcal{A}, \mathcal{TH}\}$. For a generic entity $\mathcal{X}$ in layer $\mathcal{L}$, we use $\mathcal{X} \in \mathcal{L}$ to represent an arbitrary element (e.g., $j \in \mathcal{M}$ for a mission, $i \in \mathcal{T}$ for a task). The notation $\mathcal{W}^{\mathcal{L}}_{\mathcal{X},d}$ denotes the weight of entity $\mathcal{X}$ in layer $\mathcal{L}$ for dimension $d \in \{C, I, A\}$.

To ensure comparability across layers, we nor-

malize weights within each layer using:

Weights are derived using the Risk-to-Mission Assessment Process (RiskMAP) [17] and the Analytic Hierarchy Process (AHP). Experts perform pairwise CIA comparisons ...producing the weight vector $\mathbf{w}_{CIA}$, where w_C, w_I, and w_A represent the relative weights of the Confidentiality, Integrity, and Availability dimensions, respectively:

$$\mathbf{w}_{CIA} = [w_C, w_I, w_A], \quad w_C + w_I + w_A = 1, \tag{5}$$

where w_C, w_I, and w_A represent the relative importance of confidentiality, integrity, and availability respectively. Eigenvector consistency is verified with consistency ratio CR < 0.1 to ensure judgment reliability.

Table 1 provides comprehensive definitions for all variables and notations used throughout the weight propagation formulas in this subsection.

Table 1. Notation Definitions for Weight Propagation

Notation	Definition
$\mathcal{W}^{T}_{i,d}$	Weight of task i for CIA dimension d
$\mathcal{W}^{MO}_{j,d}$	Weight of mission objective j for dimension d
$R^{MT}_{ji,d}$	Relationship strength from mission j to task i in dimension d
N_{MO}	Number of mission objectives
$\mathcal{W}^{A}_{k,d}$	Weight of asset k for dimension d
$R^{TA}_{ik,d}$	Relationship strength from task i to asset k in dimension d
N_T	Number of tasks
$\mathcal{W}^{Th}_{z,d}$	Weight of threat z for dimension d
$R^{ATh}_{kz,d}$	Relationship strength from asset k to threat z in dimension d
N_A	Number of assets
$\hat{\mathcal{W}}^{\mathcal{L}}_{\mathcal{X},d}$	Normalized weight of entity $\mathcal{X}$ in layer $\mathcal{L}$ for dimension d

As defined in Table 1, these weights form the quantitative basis for hierarchical propagation across mission ($\mathcal{M}$), task ($\mathcal{T}$), asset ($\mathcal{A}$), and threat ($\mathcal{TH}$) layers. The propagation follows a top-down approach where mission criticality flows to lower organizational layers:

$$\mathcal{W}^{T}_{i,d} = \sum_{j=1}^{N_{\text{MO}}} \mathcal{W}^{MO}_{j,d} \cdot R^{MT}_{ji,d}, \tag{6}$$

$$\mathcal{W}^{A}_{k,d} = \sum_{i=1}^{N_T} \mathcal{W}^{T}_{i,d} \cdot R^{TA}_{ik,d}, \tag{7}$$

$$\mathcal{W}^{Th}_{z,d} = \sum_{k=1}^{N_A} \mathcal{W}^{A}_{k,d} \cdot R^{ATh}_{kz,d}. \tag{8}$$

Equations (6)–(8) compute aggregated weights at each hierarchical level by multiplying parent-layer weights with corresponding relationship strengths (as defined in Table 1) and summing across all parent entities. This captures how mission priorities cascade through organizational dependencies.

The normalization process scales unadjusted weights into unit intervals to enable consistent comparison across layers:

$$\hat{\mathcal{W}}^{\mathcal{L}}_{\mathcal{X},d} = \frac{\mathcal{W}^{\mathcal{L}}_{\mathcal{X},d}}{\max_{\mathcal{X}' \in \mathcal{L}} (\mathcal{W}^{\mathcal{L}}_{\mathcal{X}',d})}, \tag{9}$$

where entity $\mathcal{X}$ belongs to layer $\mathcal{L}$, and the denominator represents the maximum weight value within that layer for dimension d (refer to Table 1 for variable definitions). This normalization, defined in Equation (9), provides numerical consistency across all hierarchical layers and CIA dimensions, ensuring that weights from different organizational levels can be meaningfully compared.

3.4 Example: Weight Propagation from Mission to Threat

To illustrate the weight propagation mechanism, consider a simplified scenario with three missions (MO_1, MO_2, MO_3), three tasks (T_1, T_2, T_3), three assets (A_1, A_2, A_3), and three threats (Th_1, Th_2, Th_3). We demonstrate the propagation for the Confidentiality dimension ($d = C$); the same process applies to Integrity and Availability.

Step 1: Mission Layer (Initial Weights)

Expert judgment assigns normalized mission weights:

$$\mathcal{W}^{MO}_{MO_1,C} = 0.500, \quad \mathcal{W}^{MO}_{MO_2,C} = 0.300,$$
$$\mathcal{W}^{MO}_{MO_3,C} = 0.200$$
$$\sum_{j=1}^{3} \mathcal{W}^{MO}_{MO_j,C} = 1.000$$

Step 2: Task Layer (Propagation and Normalization)

Relationship strengths $R^{MT}_{ji,C}$ between missions and tasks are defined (e.g., via RiskMAP dependency

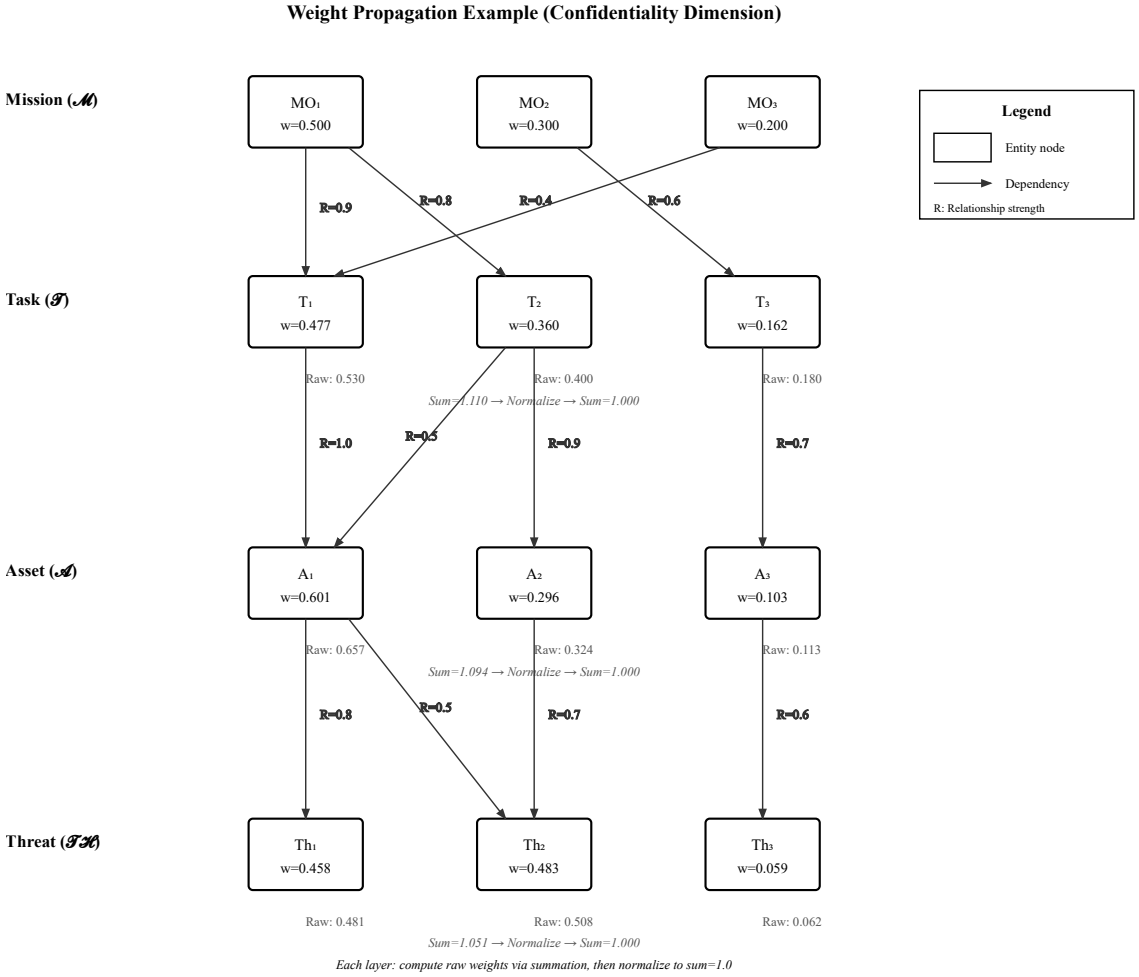

Figure 3. Weight propagation example for the Confidentiality dimension across the four-layer hierarchy (Mission → Task → Asset → Threat). Each node shows its raw weight (computed via summation) and normalized weight. Edge labels indicate relationship strengths (R values). The normalization step at each layer ensures weights sum to 1.0, preserving the hierarchical priority structure while enabling consistent threat ranking.

analysis). Using the propagation formula:

$$\mathcal{W}_{i,C}^{T} = \sum_{j=1}^{3} \mathcal{W}_{j,C}^{MO} \cdot R_{ji,C}^{MT}$$

For example, assume:

- $R_{MO_1 \to T_1,C}^{MT} = 0.9$, $R_{MO_1 \to T_2,C}^{MT} = 0.8$
- $R_{MO_2 \to T_3,C}^{MT} = 0.6$
- $R_{MO_3 \to T_1,C}^{MT} = 0.4$

Raw task weights:

$$\begin{aligned}
\mathcal{W}_{T_1,C}^{T} &= 0.500 \times 0.9 + 0.300 \times 0.0 \\
&\quad + 0.200 \times 0.4 = 0.530 \\
\mathcal{W}_{T_2,C}^{T} &= 0.500 \times 0.8 + 0.300 \times 0.0 \\
&\quad + 0.200 \times 0.0 = 0.400 \\
\mathcal{W}_{T_3,C}^{T} &= 0.500 \times 0.0 + 0.300 \times 0.6 \\
&\quad + 0.200 \times 0.0 = 0.180 \\
\text{Raw sum} &= 1.110
\end{aligned}$$

Normalized task weights:

$$\begin{aligned}
\hat{\mathcal{W}}_{T_1,C}^{T} &= \frac{0.530}{1.110} = 0.477, \\
\hat{\mathcal{W}}_{T_2,C}^{T} &= \frac{0.400}{1.110} = 0.360, \\
\hat{\mathcal{W}}_{T_3,C}^{T} &= \frac{0.180}{1.110} = 0.162 \\
\sum_{i=1}^{3} \hat{\mathcal{W}}_{T_i,C}^{T} &= 1.000
\end{aligned}$$

Step 3: Asset Layer (Propagation and Normalization)

Using task-to-asset relationships $R^{TA}_{ik,C}$ (e.g., $R^{TA}_{T_1\to A_1,C} = 1.0$, $R^{TA}_{T_2\to A_1,C} = 0.5$, $R^{TA}_{T_2\to A_2,C} = 0.9$, $R^{TA}_{T_3\to A_3,C} = 0.7$):

$$\mathcal{W}^{A}_{k,C} = \sum_{i=1}^{3} \hat{\mathcal{W}}^{T}_{i,C} \cdot R^{TA}_{ik,C}$$

Raw asset weights:

$$\begin{aligned}\mathcal{W}^{A}_{A_1,C} &= 0.477 \times 1.0 + 0.360 \times 0.5 \\ &\quad + 0.162 \times 0.0 = 0.657 \\ \mathcal{W}^{A}_{A_2,C} &= 0.477 \times 0.0 + 0.360 \times 0.9 \\ &\quad + 0.162 \times 0.0 = 0.324 \\ \mathcal{W}^{A}_{A_3,C} &= 0.477 \times 0.0 + 0.360 \times 0.0 \\ &\quad + 0.162 \times 0.7 = 0.113 \\ \text{Raw sum} &= 1.094\end{aligned}$$

Normalized asset weights:

$$\begin{aligned}&\hat{\mathcal{W}}^{A}_{A_1,C} = 0.601, \quad \hat{\mathcal{W}}^{A}_{A_2,C} = 0.296, \\ &\hat{\mathcal{W}}^{A}_{A_3,C} = 0.103 \\ &\sum_{k=1}^{3} \hat{\mathcal{W}}^{A}_{A_k,C} = 1.000\end{aligned}$$

Step 4: Threat Layer (Propagation and Normalization)

Using asset-to-threat relationships $R^{ATh}_{kz,C}$ (e.g., $R^{ATh}_{A_1\to Th_1,C} = 0.8$, $R^{ATh}_{A_1\to Th_2,C} = 0.5$, $R^{ATh}_{A_2\to Th_2,C} = 0.7$, $R^{ATh}_{A_3\to Th_3,C} = 0.6$):

$$\mathcal{W}^{Th}_{z,C} = \sum_{k=1}^{3} \hat{\mathcal{W}}^{A}_{k,C} \cdot R^{ATh}_{kz,C}$$

Raw threat weights:

$$\begin{aligned}\mathcal{W}^{Th}_{Th_1,C} &= 0.601 \times 0.8 + 0.296 \times 0.0 \\ &\quad + 0.103 \times 0.0 = 0.481 \\ \mathcal{W}^{Th}_{Th_2,C} &= 0.601 \times 0.5 + 0.296 \times 0.7 \\ &\quad + 0.103 \times 0.0 = 0.508 \\ \mathcal{W}^{Th}_{Th_3,C} &= 0.601 \times 0.0 + 0.296 \times 0.0 \\ &\quad + 0.103 \times 0.6 = 0.062 \\ \text{Raw sum} &= 1.051\end{aligned}$$

Normalized threat weights:

$$\begin{aligned}&\hat{\mathcal{W}}^{Th}_{Th_1,C} = 0.458, \quad \hat{\mathcal{W}}^{Th}_{Th_2,C} = 0.483, \\ &\hat{\mathcal{W}}^{Th}_{Th_3,C} = 0.059 \\ &\sum_{z=1}^{3} \hat{\mathcal{W}}^{Th}_{Th_z,C} = 1.000\end{aligned}$$

Interpretation

The final normalized threat weights indicate that Th_2 (weight 0.483) poses the highest confidentiality risk, followed by Th_1 (0.458) and Th_3 (0.059). This hierarchical propagation ensures that mission-level priorities cascade through tasks and assets to inform threat prioritization. The same process applies independently to Integrity and Availability dimensions, with final threat scores computed using the CIA weight vector $\mathbf{w}_{CIA} = [w_C, w_I, w_A]$.

3.5 Multi-Criteria Countermeasure Optimization

This subsection formulates the multi-objective optimization problem that balances threat mitigation, cost constraints, service quality, and lateral risk containment.

The optimization framework evaluates each countermeasure cm_j through four complementary criteria that collectively address both security effectiveness and operational feasibility. Each criterion captures a distinct aspect of countermeasure value, enabling holistic decision-making under resource constraints.

Threat Risk Mitigation (TR): This criterion quantifies the direct reduction in threat exposure achieved by implementing countermeasure cm_j. Let $TH(cm_j)$ denote the set of threats addressed by cm_j. For each threat $z \in TH(cm_j)$ and each security dimension $d \in \{C, I, A\}$, let $\hat{W}^{Th}_{z,d}$ denote the normalized threat weight and $R^{Th}_{z,d}$ denote the corresponding mitigation effectiveness. The threat mitigation score aggregates these terms as follows:

$$\text{TR}(cm_j) = \sum_{z \in \mathcal{TH}(cm_j)} \sum_{d \in \{C,I,A\}} \hat{\mathcal{W}}^{Th}_{z,d} \cdot R^{Th}_{z,d} \cdot M^{th}_{jd}, \tag{10}$$

where $R^{Th}_{z,d}$ represents the risk level of threat z in dimension d, and M^{th}_{jd} denotes the mitigation effectiveness of countermeasure cm_j against dimension d (typically derived from CVSS scores or expert assessments). Equation (10) computes the weighted sum of risk reductions across all threats and CIA dimensions. These symbols are also summarized in Table 1.

Cost Efficiency (C): This criterion normalizes countermeasure costs to enable comparison across different price ranges. Given the cost $C(cm_j)$ of countermeasure cm_j [16], the normalized cost efficiency

is:

$$\hat{C}(cm_j) = 1 - \frac{C(cm_j)}{\max_{j'} C(cm_{j'})}, \tag{11}$$

where the denominator represents the maximum countermeasure cost in the candidate set. Higher values of $\hat{C}(cm_j)$ in Equation (11) indicate more cost-effective solutions.

Quality-of-Service Impact (Q): This criterion assesses both direct and indirect operational impacts of countermeasure deployment [16]. Direct service quality (DSQ) measures immediate performance degradation on affected systems, while indirect service quality (ISQ) captures cascading effects on dependent systems. Let s_{ij} represent the dependency strength from countermeasure cm_j to system i, and q_i denote the service quality metric of system i. The direct impact is:

$$\text{DSQ}(cm_j) = \frac{\sum_i s_{ij} \cdot q_i}{\sum_i s_{ij}}, \tag{12}$$

which computes the weighted average quality across directly affected systems. Similarly, let r_{jk} represent the dependency from cm_j to downstream countermeasure cm_k. The indirect impact propagates through dependency chains:

$$\text{ISQ}(cm_j) = \frac{\sum_k r_{jk} \cdot \text{DSQ}(cm_k)}{\sum_k r_{jk}}, \tag{13}$$

where $r_{jk} > 0$ indicates that deploying cm_j triggers or affects downstream countermeasure cm_k. We assume **acyclic dependencies**: if cyclic relationships exist (e.g., $cm_A \to cm_B \to cm_A$), they are treated as a single composite countermeasure package. Dependencies **may overlap**—multiple countermeasures can share a common downstream control. In such cases, Equation 13 aggregates effects independently for each countermeasure, which may overestimate total system impact when multiple countermeasures are deployed simultaneously. This conservative approach provides an upper bound on individual countermeasure costs during the selection phase.

Example: Consider three countermeasures with dependencies:

- cm_1 (Firewall Rule Update) $\to$ cm_3 (Network Latency Increase)
- cm_2 (IDS Signature Deployment) $\to$ cm_3 (Network Latency Increase)

Both cm_1 and cm_2 induce indirect degradation through cm_3. If $r_{13} = r_{23} = 1$, then:

$$\text{ISQ}(cm_1) = \text{DSQ}(cm_3), \quad \text{ISQ}(cm_2) = \text{DSQ}(cm_3).$$

The impact of cm_3 contributes to the service quality cost of both cm_1 and cm_2 independently. While this may double-count shared effects when both are selected, it ensures that each countermeasure's worst-case impact is captured in the optimization model.

capturing secondary effects on interdependent controls. The composite QoS metric combines both components with weighting parameter $\alpha \in [0, 1]$:

$$\hat{Q}(cm_j) = 1 - \alpha \cdot \text{DSQ}(cm_j) - (1-\alpha) \cdot \text{ISQ}(cm_j), \tag{14}$$

where higher $\hat{Q}(cm_j)$ values indicate minimal service degradation. Equations (12)–(14) collectively quantify operational impact across direct and indirect system dependencies.

Lateral Risk Reduction (LRR): This criterion quantifies how effectively a countermeasure prevents threat propagation across network boundaries. Let $\mathcal{N}(z)$ denote the set of threats reachable from threat z through lateral movement paths. The lateral containment effectiveness is:

$$\widehat{\text{LR}}(cm_j) = \frac{\sum_{z \in \mathcal{TH}(cm_j)} \sum_{z' \in \mathcal{N}(z)} (R^{Th}_{z,d} - R^{Th,\text{post}}_{z',d})}{\sum_{z \in \mathcal{TH}(cm_j)} \sum_{z' \in \mathcal{N}(z)} R^{Th}_{z,d}}, \tag{15}$$

where $R^{Th,\text{post}}_{z',d}$ represents the residual risk at threat z' after implementing cm_j. Equation (15) measures the proportional reduction in lateral threat diffusion potential.

The multi-objective utility function combines these four criteria with user-defined preference weights:

$$\begin{aligned} U(cm_j) = {} & w_{TR} \cdot \text{TR}(cm_j) + w_C \cdot \hat{C}(cm_j) \\ & + w_Q \cdot \hat{Q}(cm_j) + w_{LR} \cdot \widehat{\text{LR}}(cm_j), \end{aligned} \tag{16}$$

where w_{TR}, w_C, w_Q, and w_{LR} are non-negative weights satisfying $\sum_i w_i = 1$. This formulation in Equation (16) enables flexible prioritization based on organizational security policies and operational requirements. The optimization algorithm iteratively selects countermeasures maximizing $U(cm_j)$ subject to budget constraint $\sum_{cm_j \in \mathcal{C}_{\text{selected}}} C(cm_j) \leq \mathcal{C}_{\max}$ until risk reduction saturation or resource exhaustion.

This iterative selection is implemented as a **greedy algorithm** that operates as follows: at each iteration, the countermeasure with the highest utility score $U(cm_j)$ that fits within the remaining budget is selected and added to $\mathcal{C}_{\text{selected}}$. The process continues until either risk reduction saturation is achieved or budget resources are exhausted. The computational complexity is $O(n \log n)$ for sorting countermeasures by utility, followed by $O(n)$ for the greedy selection, where n is the number of candidate countermeasures. This greedy approach pro-

vides near-optimal solutions for the multi-criteria optimization problem while maintaining efficiency suitable for real-time security response scenarios.

3.6 Post-Implementation Risk Reassessment

This subsection describes the dynamic recalculation of residual risks following countermeasure deployment, enabling continuous security posture monitoring.

After countermeasure deployment, residual risks are recomputed to reflect the new security state. For each threat z addressed by countermeasure cm_j, the updated risk level is:

$$R_{z,d}^{Th,\text{new}} = R_{z,d}^{Th}(1 - \text{TR}_z(cm_j)), \tag{17}$$

where $\text{TR}_z(cm_j)$ represents the mitigation effectiveness specific to threat z. Equation (17) applies proportional risk reduction based on countermeasure capability.

These updated threat risks propagate upward through the organizational hierarchy using the normalized weights established in Equations (6)–(8):

$$R_{k,d}^{A,\text{new}} = \sum_z R_{z,d}^{Th,\text{new}} \hat{\mathcal{W}}_{z,d}^{Th}, \tag{18}$$

$$R_{i,d}^{T,\text{new}} = \sum_k R_{k,d}^{A,\text{new}} \hat{\mathcal{W}}_{k,d}^{A}, \tag{19}$$

$$R_{j,d}^{MO,\text{new}} = \sum_i R_{i,d}^{T,\text{new}} \hat{\mathcal{W}}_{i,d}^{T}. \tag{20}$$

Equations (18)–(20) sequentially aggregate residual risks from threats to assets, tasks, and finally mission objectives, mirroring the forward propagation structure but incorporating post-mitigation risk values.

Global risk after mitigation is computed as the weighted sum of mission-level risks:

$$R_{\text{Global},d}^{\text{new}} = \sum_{j=1}^{N_{\text{MO}}} R_{j,d}^{MO,\text{new}} \hat{\mathcal{W}}_{j,d}^{MO}. \tag{21}$$

The difference $R_{\text{Global},d} - R_{\text{Global},d}^{\text{new}}$ quantifies the absolute risk reduction achieved across each CIA dimension through the implemented countermeasure portfolio.

3.7 Performance Metrics

This subsection defines quantitative indicators for evaluating the effectiveness and efficiency of the countermeasure selection strategy.

Overall effectiveness is measured through mission-level indices that capture both security improvement and resource optimization.

Mission-Level Risk Reduction Percentage (MLRRP): This metric quantifies the proportional decrease in global risk for each CIA dimension:

$$\text{MLRRP}_d = \frac{R_{\text{Global},d} - R_{\text{Global},d}^{\text{new}}}{R_{\text{Global},d}} \times 100\%. \tag{22}$$

MLRRP values approaching 100% indicate near-complete risk mitigation, while lower values suggest residual vulnerabilities requiring additional countermeasures or acceptance decisions. Equation (22) provides dimension-specific assessments enabling targeted security investments.

Resource Utilization Efficiency (RUE): This metric evaluates cost-effectiveness by measuring risk reduction per monetary unit invested:

$$\text{RUE} = \frac{\sum_d (R_{\text{Global},d} - R_{\text{Global},d}^{\text{new}})}{\sum_{cm_j \in \mathcal{C}_{\text{selected}}} C(cm_j)}, \tag{23}$$

where the numerator sums risk reductions across all CIA dimensions and the denominator represents total countermeasure expenditure. Higher RUE values signify superior cost-benefit ratios, indicating that the optimization strategy efficiently allocates limited resources to maximize security improvements. Equation (23) enables comparative analysis across different countermeasure portfolios and budget scenarios.

Collectively, Phases 1–4 establish a dynamic cyber-defense ecosystem. GMASO integrates weighted graph modeling, agent-based coordination, and multi-objective optimization to transform raw data—assets, threats, vulnerabilities, and controls—into adaptive decision support for mission-focused security enhancement. Continuous propagation and reassessment sustain an intelligent, data-driven approach to risk-aware, resource-efficient, and mission-aligned cybersecurity management.

4 Experimental Results

The proposed GMASO framework was validated within an infrastructure of the university's Informatics comprising three mission objectives, nine business tasks, and twenty-three assets spanning the CIA dimensions. The simulated environment included nineteen countermeasures addressing fifteen threats via twelve vulnerability pathways. Implementation was carried out in AnyLogic, with all artifacts, datasets, and simulation models publicly available at https://github.com/sajious/SA_System for reproducibility.

Figure 4 illustrates the institute's network architecture, highlighting hierarchical security domains and critical components. The topology highlights interdependencies among components and demonstrates how failures at boundary devices can cascade into

higher levels, forming the basis for dependency modeling and quantitative analysis of threat propagation and countermeasure effects.

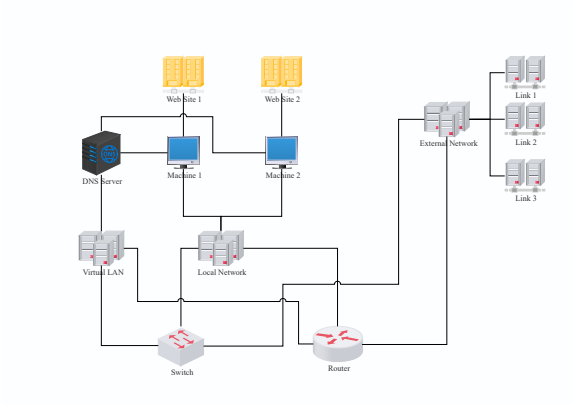

Figure 4. Informatics Network Topology Overview showing security domains, critical infrastructure components, and interconnection pathways used in the case study validation.

Table 2 summarizes key parameters guiding countermeasure selection. The budget cap was set at 5000 monetary units, and the multi-criteria weighting scheme reflected operational priorities: availability ($\gamma_A = 0.5$), integrity ($\gamma_I = 0.4$), and confidentiality ($\gamma_C = 0.1$). Secondary utility weights emphasized Threat Risk Reduction ($\omega_{TR} = 0.4$), Deployment Complexity ($\omega_C = 0.2$), Quality-of-Service Impact ($\omega_Q = 0.3$), and Lateral Risk Reduction ($\omega_{LR} = 0.1$), defining optimization constraints and objectives for cost-effective resource allocation. All weighting coefficients and parameter values listed in Table 2 were elicited from domain experts within the university's Informatics Department, based on their operational experience and assessment of asset criticality.

Table 2. Parameterization for Optimal Countermeasure Selection in the Informatics Case Study

Parameter	Value
Max Budget ($)	5000
Confidentiality Weight, γ_C	0.1
Integrity Weight, γ_I	0.4
Availability Weight, γ_A	0.5
Threat Risk Reduction Weight, ω_{TR}	0.4
Deployment Complexity Weight, ω_C	0.2
QoS Impact Weight, ω_Q	0.3
Lateral Risk Reduction Weight, ω_{LR}	0.1
Manually Selected Countermeasure	Null

The optimization produced a portfolio of five countermeasures with a total cost of 1400 units, reducing risks across all missions while preserving 72% of the budget. Nineteen candidates were scored using the utility function, revealing notable performance variations across threat contexts and underscoring the need for threat-specific effectiveness modeling.

Figure 5 presents the comparative Multi-Layer Risk Reduction Profile (MLRRP) across confidentiality, integrity, and availability. CM5 achieved balanced peak values above 47% in all dimensions; CM6–CM10 delivered strong, cost-effective performance; CM1–CM4 provided moderate, targeted protection; while CM11–CM19 exhibited lower overall impact, indicating limited applicability. This quantitative comparison supports informed prioritization aligned with mission-specific CIA requirements.

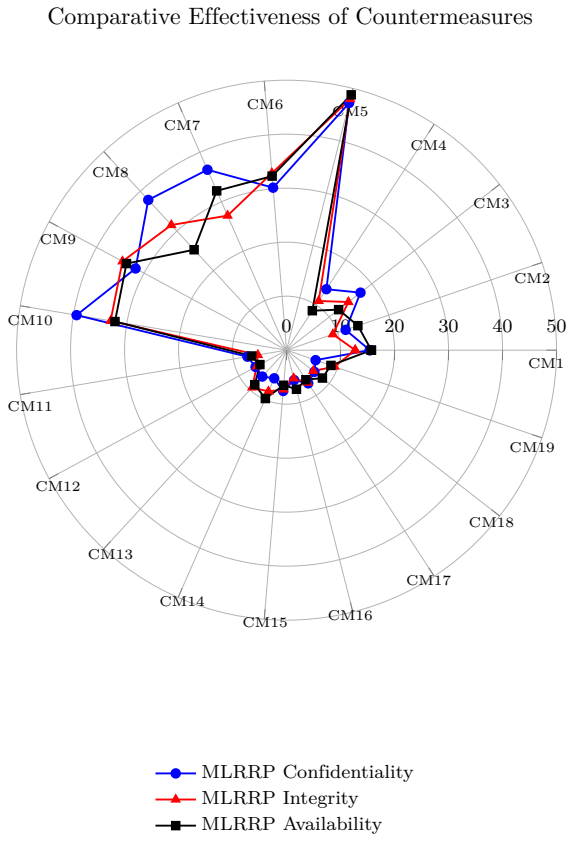

Figure 5. Radar chart showing the comparative effectiveness of countermeasures (CM1–CM19) across the MLRRP criteria of Confidentiality, Integrity, and Availability.

Figure 6 depicts the Resource Utilization Efficiency (RUE) radar comparing each countermeasure's effectiveness-to-cost ratio. CM6 achieved the highest efficiency (31.8%), followed by CM4 (11.0%) and CM10 (7.0%). High-performing controls CM5–CM10 maintained RUE between 5.8–31.8%, while CM11–CM19 remained low (0.4–6.1%). This ranking supports investment decisions based on measurable return on security expenditure, vital under constrained budgets.

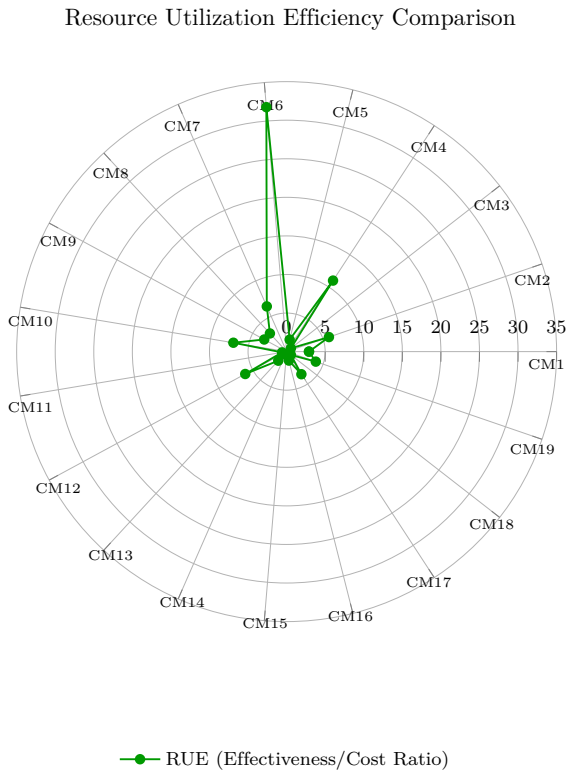

Figure 6. Radar chart showing Resource Utilization Efficiency comparison across all countermeasures (CM1–CM19), displaying normalized effectiveness-to-cost ratio metrics.

Overall, GMASO significantly improved both risk mitigation efficiency and cost precision compared to conventional static methods. The MLRRP quantifies the relative decrease in aggregated mission risk after deploying optimized countermeasures, representing the percentage improvement in overall mission assurance. In this study, mission-level risk reductions ranged between 26–39% across the three CIA dimensions — availability (28–38%), integrity (28–39%), and confidentiality (26–35%) — indicating substantial enhancement of system resilience under the defined budget constraints. Detailed evaluation datasets, risk propagation matrices, and analytical scripts are available in the supplementary repository for independent replication and further research.

4.1 Sensitivity Analysis

Since the proposed GMASO framework relies on weighted criteria in the multi-objective utility function, a sensitivity analysis was conducted to evaluate the robustness of the countermeasure selection process with respect to variations in these weights. The baseline configuration assigns weights $\omega_{TR} = 0.4$, $\omega_C = 0.2$, $\omega_Q = 0.3$, and $\omega_{LR} = 0.1$ to the four optimization criteria: threat risk mitigation (TR), cost efficiency (C), quality-of-service impact (Q), and lateral risk reduction (LR).

To assess robustness, each weight was independently varied by $\pm 10\%$ and $\pm 20\%$ while proportionally adjusting the remaining weights to maintain normalization. For each scenario, the utility score of the candidate countermeasures was recalculated and the selected optimal countermeasure was recorded.

Table 4 summarizes the results of the sensitivity analysis. As shown, the optimal countermeasure remains unchanged (CM6) across all tested scenarios, while the utility scores exhibit only moderate variations. This indicates that the proposed GMASO framework demonstrates stable decision outcomes under reasonable perturbations of the weighting parameters, supporting the robustness of the multi-criteria optimization process.

5 Conclusion

This study presented the GMASO framework, advancing cybersecurity optimization for mission-critical systems through graph-based dependency modeling and multi-agent intelligence. Empirical validation confirmed significant gains in risk mitigation efficiency, resource utilization, and mission alignment over conventional approaches.

The CIA-weighted dependency model enables granular quantification of inter-system security relationships beyond binary assessments, accurately capturing breach propagation across confidentiality, integrity, and availability dimensions. This precision supports targeted countermeasure selection and strengthens organizational resilience against cascading risks.

By incorporating threat-specific countermeasure mapping—an approach introduced in our previous mission-centric framework [1]—GMASO departs from uniform effectiveness assumptions, achieving superior resource efficiency through context-aware profiling and hierarchical multi-agent decision-making.

Dynamic risk assessment mechanisms allow continuous recalibration and adaptive countermeasure recommendations under evolving threat conditions, overcoming limitations of static security models.

Future work will address several key directions. **Refined dependency modeling** will explore graph partitioning techniques and shared-node weighting schemes to eliminate redundant summation when multiple countermeasures share common downstream controls, improving accuracy in cascading effect estimation. Additionally, **scalability optimization** for large-scale enterprise environments, **domain-specific calibration** of risk and service quality parameters, and the **integration of machine learning** for predictive threat modeling

Table 3. Comparison of GMASO with representative cybersecurity optimization approaches

Method	Approach Type	Optimization Objective	Domain	Key Feature / Result
Viduto et al. [2]	Static	Risk–cost trade-off	Enterprise network	Fixed weighting; limited adaptability
Kotenko [3]	Multi-agent	System survivability	Simulated network	Cooperative agents; moderate risk reduction (12–18%)
Shameli-Sendi [13]	Adaptive	Dynamic mitigation	Cloud environment	Real-time risk update; higher computation cost
GMASO (This work)	**Hierarchical multi-agent**	**Mission-level risk reduction**	**Informatics case study**	**26–39% MLRRP improvement; mission-aware and cost-constrained optimization**

Table 4. Sensitivity Analysis of Key Decision-Making Criteria in Selecting the Countermeasure

Scenario	ω_{TR}	ω_{LR}	ω_{TC}	ω_{QoS}	Selected CM	Utility Score
Baseline	0.400	0.200	0.300	0.100	CM6	0.849
S1 (ω_{TR} +20%)	0.480	0.173	0.260	0.087	CM6	0.862
S2 (ω_{TR} -20%)	0.320	0.227	0.340	0.113	CM6	0.835
S3 (ω_{LR} +20%)	0.384	0.240	0.288	0.088	CM6	0.851
S4 (ω_{LR} -20%)	0.416	0.160	0.312	0.112	CM6	0.846
S5 (ω_{TC} +20%)	0.375	0.188	0.360	0.077	CM6	0.838
S6 (ω_{TC} -20%)	0.424	0.212	0.240	0.124	CM6	0.859
S7 (QoS +20%)	0.392	0.196	0.292	0.120	CM6	0.852
S8 (QoS -20%)	0.408	0.204	0.308	0.080	CM6	0.845
S9 (ω_{TR} +50%)	0.600	0.133	0.200	0.067	CM6	0.870
S10 (ω_{TR} -50%)	0.200	0.267	0.400	0.133	CM6	0.828
S11 (ω_{LR} +50%)	0.360	0.300	0.240	0.100	CM6	0.854
S12 (ω_{LR} -50%)	0.440	0.100	0.340	0.120	CM6	0.843
S13 (QoS +50%)	0.380	0.190	0.280	0.150	CM6	0.855
S14 (QoS -50%)	0.420	0.210	0.320	0.050	CM6	0.841

and automated parameter tuning will enhance the framework's adaptability and operational efficiency. As cyber threats intensify, frameworks like GMASO provide essential, data-driven decision support for maximizing security effectiveness within operational constraints.

References

[1] Sajed Yousefi Mashhour, Motahareh Dehghan, Babak Sadeghian, and Alireza Hashemi Golpayegani. Mission-centric countermeasure selection in cybersecurity situation awareness systems. *ISeCure*, 2026. .

[2] V. Viduto, C. Maple, W. Huang, and D. López-Peréz. A novel risk assessment and optimisation model for a multi-objective network security countermeasure selection problem. *Decision Support Systems*, 2012.

[3] I. Kotenko and E. Doynikova. Dynamical calculation of security metrics for countermeasure selection in computer networks. In *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016.

[4] A. Shameli-Sendi and M. Dagenais. Orcef: Online response cost evaluation framework for intrusion response system. *Journal of Network and Computer Applications*, 2015.

[5] P. Nespoli, D. Papamartzivanos, F. Gómez Mármol, and G. Kambourakis. Optimal countermeasures selection against cyber attacks: A comprehensive survey on reaction frameworks. *IEEE Communications Surveys & Tutorials*, 2018.

[6] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat. A critical cybersecurity analysis and future research directions for the internet of things: A comprehensive review. *Sensors*, 2023.

[7] A. Roy, D. S. Kim, and K. S. Trivedi. Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, 2012.

[8] Kjell Hausken, Jonathan W. Welburn, and Jun Zhuang. A review of attacker–defender games and cyber security. *Games*, 2024. .

[9] S. Wang, Z. Zhang, and Y. Kadobayashi. Exploring attack graph for cost-benefit security hardening: A probabilistic approach. *Computers & Security*, 2013.

[10] R. Dewri, I. Ray, N. Poolsappasit, and D. Whitley. Optimal security hardening on attack tree models of networks: a cost-benefit analysis. *International Journal of Information Security*, 2012.

[11] G. Gonzalez-Granadillo, J. Garcia-Alfaro, E. Alvarez, M. El-Barbori, and H. Debar. Selecting optimal countermeasures for attacks against critical systems using the attack volume model and the rori index. *Computers & Electrical Engineering*, 2015.

[12] Abdelkarim Ait Temghart, Mbarek Marwan,

and Mohamed Baslam. Stackelberg security game for optimizing cybersecurity decisions in cloud computing. *Security and Communication Networks*. .

[13] A. Shameli-Sendi, H. Louafi, W. He, and M. Cheriet. Dynamic optimal countermeasure selection for intrusion response system. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[14] F. Ö. Sönmez and B. G. Kılıç. A decision support system for optimal selection of enterprise information security preventative actions. *IEEE Transactions on Network and Service Management*, 2021.

[15] Diksha Goel, Kristen Moore, Mingyu Guo, Derui Wang, Minjune Kim, and Seyit Camtepe. Optimizing cyber defense in dynamic active directories through reinforcement learning, 2024. Accepted at ESORICS 2024.

[16] F. Li, Y. Li, S. Leng, Y. Guo, K. Geng, Z. Wang, and L. Fang. Dynamic countermeasures selection for multi-path attacks. *Computers & Security*, 2020.

[17] J. Watters, S. Morrissey, D. Bodeau, and S. C. Powers. The risk-to-mission assessment process: A sensitivity analysis and an extension to treat confidentiality issues. Technical report, 2009.

Sajed Yousefi Mashhour is a Research Assistant at Amirkabir University of Technology. He earned his Master of Science in Information Technology Engineering from Amirkabir University of Technology in 2024, after completing his Bachelor of Science in Railway Mechanical Engineering from Iran University of Science and Technology in 2020. His research interests include Countermeasure Selection, Cybersecurity Situational Awareness Systems, Agent-Based Modeling, Cybersecurity, Intrusion Response Systems, Threat Modeling, and Risk Management.

Motahareh Dehghan is an Assistant Professor in the Department of Industrial and Systems Engineering at Tarbiat Modares University. She earned her Ph.D. in Information Security from Amirkabir University of Technology. Her research interests include Information Security, Cybersecurity Situational Awareness, and Emerging Technologies, focusing on developing innovative solutions to enhance Digital Security and Resilience.

Babak Sadeghian received his Ph.D. in Computer Science from University College, University of New South Wales, Australia, in 1993. Since then, he has joined as a faculty member of the Department of Computer Engineering at Amirkabir University of Technology, Tehran, Iran. His research interests include all aspects of Information Security. His current research interests include Intrusion Detection Systems, Cybersecurity Situational Awareness, Threat Hunting, Privacy Issues, Digital Forensics and Vulnerability Analysis.

Alireza Hashemi Golpayegany is an Assistant Professor in the Department of Computer Engineering at Amirkabir University of Technology, Tehran, Iran. His research interests include financial fraud detection, blockchain security, social commerce, business process mining, and recommender systems, with particular focus on developing intelligent systems for enhancing security and trust in digital platforms.