Computers & Security 148 (2025) 104153

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Dealing with uncertainty in cybersecurity decision support

Yunxiao Zhang [a,*], Pasquale Malacaria [b]

[a] University of Exeter, United Kingdom
[b] Queen Mary University of London, United Kingdom

ARTICLE INFO

Keywords:
Robust optimization
Decision support
Uncertainty
Cyber-security
Stackelberg games
Security games
Attack graphs

ABSTRACT

The mathematical modeling of cybersecurity decision-making heavily relies on cybersecurity metrics. However, achieving precision in these metrics is notoriously challenging, and their inaccuracies can significantly influence model outcomes. This paper explores resilience to uncertainties in the effectiveness of security controls. We employ probabilistic attack graphs to model threats and introduce two resilient models: minmax regret and min-product of risks, comparing their performance.

Building on previous Stackelberg game models for cybersecurity, our approach leverages totally unimodular matrices and linear programming (LP) duality to provide efficient solutions. While minmax regret is a well-known approach in robust optimization, our extensive simulations indicate that, in this context, the lesser-known min-product of risks offers superior resilience.

To demonstrate the practical utility and robustness of our framework, we include a multi-dimensional decision support case study focused on home IoT cybersecurity investments, highlighting specific insights and outcomes. This study illustrates the framework's effectiveness in real-world settings.

1. Introduction

As technology advances, so do the threats that come with it, making the need for accurate data and statistics on cybersecurity more critical than ever. However, obtaining reliable data in this domain is challenging (Pendleton et al., 2016; Verendel, 2009; Fielder et al., 2018; Rass et al., 2015). Many organizations are reluctant to disclose information about security breaches due to concerns about reputational damage and legal repercussions. Also, cyber-attacks are continuously evolving, making it difficult to categorize and quantify them accurately (Banga, 2020). New types of malware and attack vectors regularly emerge, making existing data quickly outdated. There is no universally accepted standard for reporting cyber incidents, and different countries have different laws and regulations regarding cybersecurity, making international data aggregation a complex task.

One of the biggest challenges in cybersecurity is quantifying the security risk and effectiveness of security controls. Cybersecurity often deals with unknown quantities, unlike other fields where performance metrics are straightforward to define and measure. For instance, how does one measure a firewall's effectiveness that has never been breached? Is it 100% effective, or has it simply not been tested by a sophisticated enough attack? The absence of incidents does not necessarily indicate effectiveness, creating a paradox in measurement. On the whole, cybersecurity metrics rely heavily on expert judgment. These challenges are well known and considered among the most important in cybersecurity, as illustrated by the recent UK's National Cyber Security Centre (NCSC) research problems book (The National Cyber Security Centre, 2023) where "How do we create and adopt meaningful measures of cyber security?" is one of the five big problems. There is no agreed answer to the above question and relatively little work on the topic; this work contributes to this research area.

Cybersecurity metrics are extensively used in mathematical models of cybersecurity investment (Cavusoglu et al., 2008; Fielder et al., 2016; Gordon and Loeb, 2002; Khouzani et al., 2019). These models aim to support organizational cybersecurity decision-making by indicating which set of security controls is optimal in the sense of reducing the security risk against a specific threat scenario.

In this work, we accept that these metrics are intrinsically imprecise, and we therefore focus on developing cybersecurity models that are resilient to such imprecisions. We consider two possible models for resilience: the first, minmax regret, is a powerful tool from robust optimization (Ben-Tal et al., 2009; Pita et al., 2012) for managing risk and uncertainty. Regret measures the ratio or difference between the chosen security portfolio's risk and the lowest achievable risk in a scenario. Given a set of scenarios (i.e. possible values for the effectiveness of controls), minmax regret will compute the set of controls x^* of minimal maximal regret across all possible scenarios.

The second model is the min-product of security risks: in this model, the portfolio chosen is the one that minimizes the product of

* Corresponding author.
E-mail addresses: y.zhang12@exeter.ac.uk (Y. Zhang), p.malacaria@qmul.ac.uk (P. Malacaria).

https://doi.org/10.1016/j.cose.2024.104153
Received 10 June 2024; Received in revised form 23 July 2024; Accepted 7 October 2024
Available online 9 October 2024

Table 1
Main symbols used in the paper and their description.

Symbol	Description
G, $\mathcal{V}$, $\mathcal{E}$	Attack graph, set of nodes in attack graph, set of edges in attack graph
s, t	Source and target nodes in attack graph
$\overline{h}(e)$, $\underline{t}(e)$	Head and tail nodes of edge e
$C, C(e), \mathcal{L}(c)$	Set of security controls, set of controls for edge e, levels of control c
x, x_{cl}	Security portfolio, indicator variable for control c at level l
$p_e(x)$	Probability of a successful attack step associated with edge e given x
π_e	Probability of a successful attack step when e is unprotected
B_D, B_I	Budgets for direct and indirect costs
$\text{Cost}_{cl}, \text{InCost}_{cl}$	Direct and indirect costs of control c at level l
p_{ecl}, (p^g_{ecl})	Effectiveness of control c at level l on edge e (in scenario g)
$\overline{p}_{ecl}$ $(\underline{p}_{ecl})$	Upper (lower) bound of the control effectiveness interval
$\omega_{s\to t}$	A path from the source node s to the target node t
λ (λ^g)	Dual variables (in scenario g) of the maximization problem in (4)
$\mathcal{G}, \overline{\mathcal{G}}, g$	Set of scenarios, sampled set of scenarios, a scenario
x^*_g	An optimal security portfolio in scenario g
$r_g(x), r^*_g$	Security risk with security portfolio x in scenario g, minimal security risk in scenario g

risks across all scenarios. Comparing the min-product with minmax regret notices that the latter offers worst-case guarantees, but in many cases, it may be over-pessimistic compared to the former: The key takeaway from this work is that min-product provides a more balanced optimization that reduces the overall risk across all scenarios.

1.1. Outline of the paper and contributions

This work contributes to the general area of mathematical modeling of cybersecurity. In particular, it contributes to addressing questions about the resilience of such models and their extension with robust optimization techniques.

Section 2 introduces the attack graphs used to model threat scenarios and the optimization which is used as a basis for the minmax regret and min-product.

Section 3 contains the main contributions, i.e., minmax regret and min-product optimization. Starting with developing the minmax regret framework, it is shown how the original bi-level minmax regret problem can be converted to an efficient single-level MILP (mixed integer linear programming).

We then introduce min-product: To the best of our knowledge, this is the first use of this kind of optimization in this decision support context. Technically, we derive it as a modification of a minmax regret constraint, inheriting the efficient MILP properties previously derived for minmax regret. The non-linearity of the product is addressed with the standard sum-log-exp conversion.

Section 4 reports on experiments comparing minmax regret, min-product, and other possible defensive strategies. The experiments show that min-product provides the highest security (measured as average security risk over all scenarios). The section also reports on further experiments about the scalability of min-product and minmax regret, showing that min-product offers in general, better time performance than minmax regret.

Section 5 illustrates how the min-product can be used in multi-dimensional cybersecurity decision-making contexts: Here, it is used not only to select an optimal set of controls but also for more complex decisions involving different investment options. As an example of this modeling, a case study based on choosing IoT home bundles is provided. To the best of our knowledge, this is the first application of this kind of optimization to this decision-making context.

1.2. List of symbols

See Table 1.

2. Background

2.1. Related work

Security games are a special class of game theory problems for addressing security challenge (Korzhyk et al., 2011; Pita et al., 2008; Yin and Tambe, 2012; Fang et al., 2016; Paruchuri et al., 2008), in particular extensively used in cybersecurity (Pita et al., 2008; Yin and Tambe, 2012; Fang et al., 2016; Paruchuri et al., 2008; Zhang and Malacaria, 2021a; La et al., 2016; Żychowski and Mańdziuk, 2021a; Zhang and Malacaria, 2021b; Fielder et al., 2016; Durkota et al., 2015; Sawik, 2013; Zhang et al., 2023; Zhang and Malacaria, 2023; Khouzani et al., 2016, 2019). These games are typically Stackelberg games: The defender acts as the leader, anticipating the attacker's response and committing to a defence strategy. Attackers (the follower in the game) can usually observe this strategy and respond accordingly.

In a Stackelberg game, the attacker is often assumed to be rational and knowledgeable of the defender's strategy. This represents a worst-case scenario where rational attackers optimally respond to the underlying defensive strategy to maximize their payoffs. However, attackers can exhibit bounded rationality in many real-world applications, having limited observation or acting irrationally. While such irrational attackers typically result in suboptimal payoffs, they could degrade the underlying defence performance depending on the problem setting and solutions. Work in Pita et al. (2009, 2012) provides robust approaches to address bounded rational attackers, and work in Żychowski and Mańdziuk (2021b) and Yang et al. (2011) studies learning-based models for the interactions between attackers and defenders. In particular, Zhang and Malacaria (2023) provides an efficient solution for security investment games with bounded rational attackers.

Cybersecurity investment problems have been studied in several papers. One initial work is Gordon and Loeb (2002), which considers the costs and benefits of determining the cybersecurity investment. The authors in Cavusoglu et al. (2008) apply a game-theoretical framework to evaluate a firm's IT security investment levels and compare this with a decision theory approach. Subsequent research in Sawik (2013) applies financial engineering tools to IT security planning and extends this to optimizing safeguards for Industry 4.0 supply chains in Sawik (2022). Recently, the work in Abdallah et al. (2021) studies human biases in security investment decision-making. The study in Uuganbayar et al. (2021) offers a cost-efficient strategy for allocating cyber security investments, providing an exact algorithm for selecting optimal security controls and comparing it with other methods. In addition, various studies have investigated both theoretical and practical aspects of cybersecurity investment (Chronopoulos et al., 2017; Khouzani et al., 2016; Fielder et al., 2016; Smeraldi and Malacaria, 2014; Scott et al., 2022; Tsiodra et al., 2023; Zhang et al., 2023).

The closest work to this paper is Khouzani et al. (2019), which models the cybersecurity investment problem as a multi-objective bi-level Stackelberg game, using a probabilistic attack graph to represent an organization's security risk. The optimization objective is to minimize the security risk by selecting an optimal portfolio of security controls with specified security budgets. Subsequent research in Zhang and Malacaria (2021b,a, 2023) extends the efficient solution in Khouzani et al. (2019). These works integrate a Markov chain with a probabilistic attack graph to model the cybersecurity resilience of an organization, introduce an efficient Bayesian Stackelberg game when the defender is uncertain about the attacker's position within the organization, and focus on improving security against bounded rational attackers using residual budgets, respectively.

The cybersecurity problem investigated in Khouzani et al. (2019) and the subsequent work is a generalization of *network interdiction problems* (Letchford and Vorobeychik, 2013; Nandi et al., 2016; Bhuiyan et al., 2016; Smith and Song, 2020), which also belong to Stackelberg games. While most interdiction solutions apply absolute edge removal to stop the adversary's actions, it is not always feasible to eliminate a vulnerability in cybersecurity problems. For example, one cannot simply remove a router due to its potential security vulnerability without disrupting an organization's operations. Hence, Khouzani et al. (2019) uses the probabilistic attack graphs, where the effectiveness of underlying security controls is represented as a probability.

The solutions in Khouzani et al. (2019) and Zhang and Malacaria (2021b,a, 2023) depend on the placement of controls within the organization (attack graph topology), their effectiveness in mitigating security risks (cybersecurity metrics), and the rationality of attackers. Control placement within an organization is deterministic, and the rationality of attackers has been extensively addressed in the aforementioned works; however, we often lack statistical data or quantitative analysis to determine precise security metrics (Pendleton et al., 2016). In other words, the defender needs to make decisions with uncertainties of control effectiveness.

To address uncertainties in decision-making, we explore theoretical solutions to make existing strategies robust. First, the optimization can model control effectiveness as an interval coefficient, with the actual value lying within closed intervals (Steuer, 1981; Charnes et al., 1977; Ahmad et al., 2013; Wu, 2008). Next, in particular, we adopt the scenario-based approach in Kouvelis and Yu (2013). Each scenario represents a specific realization of control effectiveness values. The regret for a security portfolio in a scenario is the ratio between the security risk from the selected portfolio and the optimal risk the defender could have attained with prior knowledge of the scenario's actual control effectiveness. Then, the *relative robust decision* is the security portfolio that can minimize the maximal regret for all possible scenarios, namely minmax regret. Work in Kiekintveld et al. (2013) uses similar conceptual approaches for handling uncertainty in security games and also considers the distributional uncertainty.

To the best of our knowledge, there is no related work that uses min-product.

Previous cybersecurity works (Fielder et al., 2018; Rass et al., 2015) have investigated the natural difficulty of quantifying security risk. However those works do not consider minmax regret nor min-product of risks; moreover, they are based on "single-steps" game matrices, while our approach deals with more complex multi-stage attacks.

2.2. Attack model

We use probabilistic attack graphs to model an organization's security risk, as denoted in Khouzani et al. (2019). Such a graph is a directed multi-graph, represented as $G = \{\mathcal{V}, \mathcal{E}, \overline{h}, \underline{t}, p_e, s, t\}$. Here, $\mathcal{V}$ and $\mathcal{E}$ define nodes and edges, respectively. Nodes are an attacker's privilege state, while edges represent exploitable vulnerabilities that allow the attacker to change its privilege states. A multi-stage attack is modeled as a path from source node s to the target node t. An edge e is directed from node i to node j, with functions $\overline{h}(e)$ and $\underline{t}(e)$ return the head node and tails of edge e. Function p_e returns the attacker's success probability of exploitation on edge e, which is determined by both the attacker's baseline success rate π_e and the control effectiveness.

The defender aims to find a security portfolio to minimize the organization's security risk within budget constraints. Such a security portfolio can be expressed using binary indicators x_{cl} as follows:

$$x_{cl} \in \{0,1\}, \forall c \in C, l \in \mathcal{L}(c); \quad \sum_{l \in \mathcal{L}(c)} x_{cl} \leq 1, \forall c \in C, \tag{1}$$

where C and $\mathcal{L}(c)$ denote the set of controls and the set of intensity levels of control c. If $x_{cl} = 1$, control c at intensity level l is selected into the security portfolio; otherwise $x_{cl} = 0$. In addition, for control c, the defender can only select one intensity level, i.e., making the sum of levels for control c no greater than 1. Moreover, a control can be effective on multiple edges, and an edge may be affected by multiple controls.

Defenders can "purchase" controls to reduce the attacker's probability of success. Each control at each level has specific costs and an effectiveness coefficient.

We follow the model (Khouzani et al., 2019) for the costs to include *direct cost* Cost_{cl}, *indirect cost* InCost_{cl}. *Direct cost* represents the monetary investment in security controls, and *indirect cost* represents the indirect (negative) costs or side-effects on normal operations. Budgets are denoted as B_D and B_I.

In addition, p_{ecl} denotes a control effectiveness coefficient. For example, if control c at level l is applied with $p_{ecl} = 0.5$, the attacker's success probability of exploiting on edge e is halved. Given a security portfolio x, the overall success probability of the attacker on edge e is expressed as follows:

$$p_e(x) = \pi_e \prod_{c \in C(e), l \in \mathcal{L}(c)} (p_{ecl} x_{cl} + (1 - x_{cl})), \tag{2}$$

where $C(e)$ is the subset of controls affecting the vulnerability associated with edge e.

2.3. Optimal defensive strategy

Recall the defender's objective is to find the optimal security portfolio to minimize the organization's security risk, defined as an attacker's highest success probability. We model the interactions between defenders and attackers as a Stackelberg security game (Korzhyk et al., 2011). In this game, defenders act as leaders who anticipate the attacker's best response and commit to the optimal security portfolio that minimizes the risk. Subsequently, as followers, the attacker's best response is to choose a path that maximizes their success rate based on the implemented security portfolio. Deviation from this optimal response could result in a lower success rate for the attacker. Formally, the game setting leads to the following optimization problem:

$$\min_x r,$$

s.t.: (1),

$$\sum_{c \in C, l \in \mathcal{L}(c)} x_{cl} \cdot \text{Cost}_{cl} \leq B_D; \quad \sum_{c \in C, l \in \mathcal{L}(c)} x_{cl} \cdot \text{InCost}_{cl} \leq B_I, \tag{3}$$

$$r \geq \max_{\omega_{s \to t}} \prod_{e \in \omega_{s \to t}} p_e(x), \tag{4}$$

where $\omega_{s \to t}$ represents a complete path from the source to the target. Maximization in (4) represents the attacker's optimal response to find an attack path that maximizes its success rate. Constraints (3) ensure that the total direct costs and indirect costs do not exceed budgets B_D and B_I, respectively.

This optimization is bi-level non-linear, which is NP-hard to solve (Sinha et al., 2017). Here, we provide a high-level summary of how this optimization is solved in Khouzani et al. (2019). Given that a logarithm function $\log(x)$ is strictly monotone for $x > 0$, we can transform the maximization objective in (4) from a product to a sum,

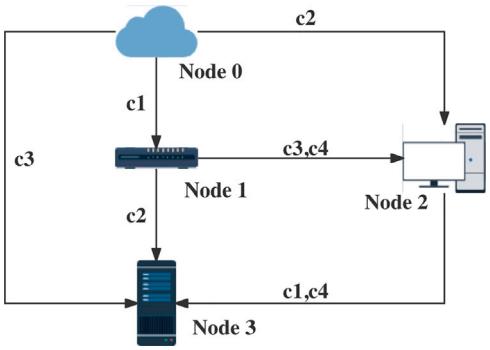


Fig. 1. Example of attack graph.

i.e., $\log(\prod_{e\in\omega_{s\to t}} p_e(x)) = \sum_{e\in\omega_{s\to t}} \log(p_e(x))$. Moreover, the path $\omega_{s\to t}$ can be mapped into new binary variables, each indicating if the attacker includes the corresponding edge in their attack path. The additional flow conservation constraints ensure the selected edges must form a complete path. This converts the non-linear maximization problem in (4) into an ILP (integer linear programming) problem. In addition, due to Lemma 2 on totally unimodular matrices in Khouzani et al. (2019), these binary variables can be relaxed to real numbers between 0 and 1. As a result, the ILP can be exactly relaxed into an LP problem.

Finally, we can dualize the original maximization to a minimization because of LP's exact relaxation and strong duality. As a result, the original optimization is transformed into a tractable MILP:

$$\begin{aligned} &\min_{x,\lambda} \lambda_s - \lambda_t, \\ &\text{s.t.: } \lambda_{\underline{t}(e)} - \lambda_{\overline{h}(e)} \geq \log(\pi_e) + \sum_{c\in C(e)} x_{cl} \log(p_{ecl}), \forall e \in \mathcal{E}, \\ &\quad (1), (3), \end{aligned} \tag{5}$$

where λ are dual variables of the maximization problem in (4).

This approach has been shown to be efficient (Khouzani et al., 2019). In particular, optimization on a random attack graph consisting of 20,000 nodes (i.e., 50,000 edges on average) and 37 controls can be solved in under four minutes.

2.4. Example

To demonstrate the concepts of optimal security defensive strategies discussed above, let us consider the example in Fig. 1. Node 0 represents the source of attacks, e.g., the external network, while Node 3 represents the target, e.g., a database. Nodes 1 and 2 are two privileged states: for example, a router and a workstation, respectively. An attacker can directly exploit the database. Alternatively, the attacker could first take control of the workstation or the router to establish a foothold within the organization and then escalate to the database via the foothold. Controls c_1 to c_4 are countermeasures to mitigate the security risk of the corresponding attack steps (edges). They could be patch management, access controls, education against social engineering attacks, firewalls, etc.

We assume each control costs one and has one intensity level. The available budget is two. We assume the effectiveness of controls c_1 to c_4 is 0.4, 0.5, 0.2, 0.1, and the baseline probabilities of edges is 1.

The optimal defender then returns the security portfolio c_1, c_3 with a security risk of 0.4.

The attacker's probability of success for each path is as follows: path $0 \to 3$ has a probability of 0.2, path $0 \to 1 \to 3$ has 0.4, path $0 \to 1 \to 2 \to 3$ has 0.032, and path $0 \to 2 \to 3$ has 0.4. We observe that the paths $0 \to 1 \to 3$ and $0 \to 2 \to 3$ have the highest probabilities of a successful attack, which represents the organization's *security risk*. We use the term *weakest path* to denote the path with the highest probability of success.

Here is the decision-making process: the optimal defender should select control c_3 to protect path $0 \to 3$. Without c_3, the attacker's success probability would be 1, resulting in a security risk of 1. Next, one more control can be added from c_1, c_2, or c_4. While c_4 is effective, its selection would leave path $0 \to 1 \to 3$ unprotected, again resulting in a security risk of 1. Therefore, the final choice should be between c_1 and c_2. Control c_1 provides a lower security risk compared to control c_2. Hence, the optimal security portfolio is c_1, c_3.

While the decision-making in the example is quite simple, the complexity increases with the number of controls and the size of the attack graph.

2.5. Extensions

For more comprehensive modeling, several extensions are incorporated into the optimal defender model (Khouzani et al., 2019). First, the model can accommodate non-independent controls by introducing a "combination control" with customized effectiveness. For instance, if controls c_a and c_b are dependent, a combination control c_{ab} can be introduced, with specific combined effectiveness and an additional constraint $x_a + x_b + x_{ab} \leq 1$.

Moreover, the model can have multiple targets by adding a "sink node" for these targets. The model can also extend to accommodate various measurements, such as the product of impact and probability of success across multiple targets. This can be achieved by assigning suitable weights to the edges connecting the target nodes to the sink node.

Several follow-up studies were developed based on this model. First, Zhang and Malacaria (2021b) combines the model with a Markov chain to evaluate system resilience against repeated attacks. Next, Zhang and Malacaria (2021a) extends the Stackelberg game model to a Bayesian setting that allows the defence strategy to consider uncertainty about the attacker's state. More recently, Zhang and Malacaria (2023) extends the model by introducing a strategy for spending residual budgets on less-protected paths. A decision-making tool (Zhang et al., 2023) has also been developed based on this approach in the context of smart home cybersecurity.

3. Modeling uncertainty

In this section, we consider two approaches to deal with uncertainty: Minmax regret and min-product.

Minmax regret is often used in robust optimization for decision-making under uncertainty. The approach aims to minimize the largest regret associated with the decision across all scenarios.

With min-product, we propose a new approach to deal with uncertainty: Instead of minimizing the largest regret, the min-product approach aims to minimize the product of risks across all scenarios.

We model a control effectiveness as a random number within a sub-interval of [0,1]. For example, when we say a control's effectiveness is between 0.4 and 0.6, we mean it can take any value between 0.4 and 0.6. The amount of uncertainty is the distance of the bounds of the interval from the mid-point of the interval. For example, if the mid-point is 0.5 and the uncertainty is 20%, we are referring to the interval $[0.4, 0.6]$.

Formally for control c at level l, we assume that the effectiveness of the control on edge e is a random number within an interval: $p_{ecl} \in [\underline{p}_{ecl}, \overline{p}_{ecl}]$. We define a scenario as a map g, where g selects a point in each interval associated with each control. We denote the set of all possible scenarios by $\mathcal{G}$.

3.1. Minmax regret

In minmax regret, the optimization aims to find a security portfolio that returns the minimal largest regret (i.e., minmax regret) across all

possible scenarios in $\mathcal{G}$. Technically, the approach followed in this work combines the framework from Khouzani et al. (2019) with the robust mathematical programming framework (Kouvelis and Yu, 2013).

Minmax Regret criterion. Given a scenario and a portfolio, the *regret* is defined as the ratio between the security risk from the selected portfolio and the optimal risk the defender could have achieved with prior knowledge of the scenario's actual effectiveness. The security portfolio that minimizes the largest regret among all scenarios is referred to as *relative robust decision* (Kouvelis and Yu, 2013).[1]

The problem is formulated as follows:

$$\arg\min_{x} \max_{g\in\mathcal{G}} r_g(x)/r_g(x_g^*) \tag{6}$$

where x_g^* denotes the optimal security portfolio for scenario g. Notice that, in general, the number of scenarios is infinite; however, we will use sampling techniques to generate a finite number of scenarios, denoted as $\overline{\mathcal{G}}$.

3.2. Converting minmax regret to a MILP

The bi-level problem in (6) can be expressed as a single-level problem by introducing an auxiliary real number, α:

$$\begin{aligned} &\min_{x,\alpha} \alpha, \\ &\text{s.t.: } \alpha \geq r_g(x)/r_g^*,\ \forall g \in \overline{\mathcal{G}}, \\ &\quad (1), (3), (4), \end{aligned} \tag{7}$$

where r_g^* represents the minimal security risk using the optimal security portfolio x_g^* for scenario g. Note that r_g^* are constants that can be computed in advance before the optimization process.

We can convert the above problem to a MILP. First, let $\beta = \log(\alpha)$. Notice then that (7) is equivalent to $\beta = \log(\alpha) \geq \log(r_g(x)) - \log(r_g^*)$. Moreover, recall that the attacker's maximization problem in (4) can be relaxed to

$$\log(r_g(x)) \geq \lambda_s^g - \lambda_t^g, \tag{8}$$

subject to

$$\lambda_{\underline{t}(e)}^g - \lambda_{\overline{h}(e)}^g \geq \log(\pi_e^g) + \sum_{c\in C(e)} x_{cl} \log(p_{ecl}^g), \forall e \in \mathcal{E}. \tag{9}$$

This is based on the relaxation presented in Section 2.3. In particular, the optimization variables λ^g are the dual variables of the attacker's maximization problem to find the weakest path given the scenario g.

Hence, inequality $\beta \geq \log(r_g(x)) - \log(r_g^*)$ is equivalent to

$$\beta + \log(r_g^*) \geq \lambda_s^g - \lambda_t^g,\ \forall g \tag{10}$$

subject to (9). Moreover, the optimizations $\min_x \alpha$ and $\min_x \log(\alpha)$ will return the same security portfolio, given that the log() function is monotonic. Hence, the minmax regret problem can be formulated as the following MILP:

$$\begin{aligned} &\min_{x} \beta, \\ &\text{s.t.: } \{(10),\ (9)\}\ \forall g \in \overline{\mathcal{G}}, (1), (3). \end{aligned}$$

3.3. Examples of minmax regret

Let us consider the simple graph with 4 nodes and four edges in Fig. 2: where the four controls c_1, c_2, c_3, c_4 have effectiveness $0.4, 0.3, 0.3, 0.4$. Then, with an uncertainty of 10% the minmax regret solution is c_2, c_3 and the regret is 1, indicating that in all scenarios, the solution will always be c_2, c_3. However, with an uncertainty of 50% there will be scenarios where the optimal solution will select c_2, c_3 (e.g., scenarios where all controls are at their upper bounds) and scenarios where the optimal solution will select c_1, c_4, e.g., with scenario $0.2, 0.45, 0.45, 0.2$. Nonetheless, the minmax regret solution will still be c_2, c_3 as it has lower maximal regret than c_1, c_4. The minmax regret in this case is 2.25, reflecting the fact that in some scenario c_2, c_3 is not optimal. In particular, the 2.25 comes from the fact that in the scenario $0.2, 0.45, 0.45, 0.2$ the minmax regret solution c_2, c_3 security risk is 0.45 while the security risk for c_1, c_4 is 0.2, and 0.45/0.2=2.25. Notice also how the solutions c_1, c_3 or c_2, c_4 are topologically impossible because they would leave some path totally undefended.

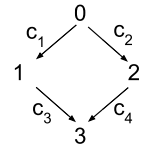


Fig. 2. A simple attack graph.

As a second example, let us consider again the attack graph in Fig. 1 from Section 2.4. At 50% uncertainty, the minmax regret with budget 2 will select the same portfolio c_1, c_3 as the optimization in Section 2.4. However, in some scenarios, the optimal solution is c_1, c_3, whereas in other scenarios, the optimal solution is c_2, c_3. This is reflected by the fact that the minmax regret when uncertainty is 50% is 2.4.

Notice that the location of controls in the graph plays a crucial role, which in many cases is more important than their effectiveness. For example, in the attack graph in Fig. 1 control c_3 has to always be chosen because it is the only one defending path $0 \to 3$, and if the budget is 2, the topology will restrict the choice of the second control to be c_1 or c_2. At budget 2, control c_4 is never selected, because c_4 leaves path $0 \to 1 \to 3$ undefended.

Topological considerations over the attack graph are key to understanding the robustness of our framework.

3.4. Min-product of security risks

Let us revisit the crucial constraint (10) in the minmax regret, i.e.

$$\beta + \log(r_g^*) \geq \lambda_s^g - \lambda_t^g,\quad \forall g.$$

If we ignore the term $\log(r_g^*)$ (which is the regret term) and we sum the right-hand side over all possible scenarios we get the following constraint:

$$\beta \geq \sum_{g\in\mathcal{G}} \lambda_s^g - \lambda_t^g, \tag{11}$$

With this change, the optimization will compute a security portfolio that minimizes the product of risks over all scenarios.

To see why, recall that in all optimizations seen so far, the objective function is the log of the function we seek. Hence, looking at the objective function implied by (11) as an exponent, we have:

$$\min_x \exp(\beta) = \min_x \exp(\sum_{g\in\mathcal{G}} \lambda_s^g - \lambda_t^g). \tag{12}$$

From Khouzani et al. (2019), the objective function $\min_x \lambda_s^g - \lambda_t^g$ is equivalent to $\min_x \log(r_g(x))$ for all g. Therefore,

$$\min_x \exp(\beta) = \min_x \exp(\sum_{g\in\mathcal{G}} \log(r_g(x))) = \min_x \prod_{g\in\mathcal{G}} r_g(x) \tag{13}$$

As a result, the min-product optimization is formally formulated as follows:

$$\begin{aligned} &\min_{x} \sum_{g\in\overline{\mathcal{G}}} \lambda_s^g - \lambda_t^g, \\ &\text{s.t.: } (9)\ \forall g \in \overline{\mathcal{G}}, (1), (3). \end{aligned}$$

[1] If the regret is based on difference rather than ratio, it is the *robust deviation* decision.

Remark 1. In order to understand the min-product approach it may be helpful to draw an analogy with Maximum Likelihood Estimation (MLE).

In MLE, we look for the model parameters that best fit the data, where "best fit" means that the product (over the data) of the conditional probability prob(data | parameter) is the highest. This means that that model parameter makes the observed data most likely.

In the min-product case, the portfolio solution is the one that best fits the product of optimal risks (over the scenarios), i.e. it is the closest to the optimal product of risks across the scenarios.

In contrast with the "worst-case analysis" of minmax regret, the product portfolio does not provide "worst-case guarantees"; however, the experiments in Section 4 show it provides a more resilient portfolio in statistical terms.

It is also worth comparing the minimization of the product of risk with the minimization of the expectation of risks. One first observation is that the product of risks is more sensitive to the higher risks than the expectation; hence, from a security point of view, as we want to prioritize the mitigation of the highest risks, minimizing the product of risk makes more sense. A further observation is computational: to the best of our knowledge, in this setting, the minimization of the expectation of risks is conic programming (Zhang and Malacaria, 2021a). Hence, it does not scale to the graph sizes considered in this work.

3.5. Mid-point portfolio

A simple way to deal with uncertainty in the effectiveness of controls is to consider the optimization from Section 2.3 where the effectiveness of controls is given by the mid-point of the uncertainty intervals.

This approach is well-known in the literature; an interesting result about this approach is in Kasperski and Zieliński (2006), where it is proven that the regret of a solution using the mid-point of uncertainty intervals is bounded by two times any solution's regret. In our setting, it is easy to show that the regret of the mid-point solution is indeed bounded, but it may not be two times any solution's regret. As discussed in the sensitivity analysis in Khouzani et al. (2019), the mid-point solution also demonstrates some level of resilience against parameter uncertainties. In the experiments, we will use the mid-point solution as the benchmark to compare the performance of other approaches.

3.5.1. Comparing mid-point portfolio with minmax regret and min-product

To illustrate the difference between mid-point portfolio and minmax regret, let us consider again the example shown in Fig. 1. Let us suppose that the mid-point effectiveness of controls c_1 to c_4 is 0.6, 0.7, 0.65, 0.5. We assume the budget allows the defender to select at most three controls.

The defender from Khouzani et al. (2019) returns the security portfolio c_1, c_3, resulting in a security risk of 0.65. Control c_3 has to be included in the security portfolio, otherwise, path $0 \rightarrow 3$ would be left unprotected. Control c_1 protects the remaining paths. Replacing control c_1 with control c_4 would leave path $0 \rightarrow 1 \rightarrow 3$ unprotected. Replacing control c_1 with control c_2 would increase the security risk to 0.7. Adding control c_2 to c_4 to the security portfolio cannot further reduce the security risk; therefore, they are not included.

Let us assume the uncertainty of control effectiveness is 30%; hence, the effectiveness of control is in a range: $p_{c_1} \in [0.42, 0.78]$, $p_{c_2} \in [0.49, 0.91]$, $p_{c_3} \in [0.455, 0.845]$, and $p_{c_4} \in [0.35, 0.65]$.

Both the minmax regret and min-product defenders select the security portfolio c_1, c_2, c_3, which provide more robust protection than the mid-point portfolio c_1 and c_3.

For example, if we take the scenario where the control effectiveness of c_1 through c_4 is 0.78, 0.49, 0.455, and 0.35, then the security portfolio c_1, c_3 returns a security risk of 0.78, whereas the security portfolio c_1, c_2, c_3 returns a security risk of 0.455.

4. Experiments

This section reports experiments comparing various defensive strategies, including minmax regret and min-product. The results indicate that min-product offers the highest security, as measured by the average security risk across all scenarios. Additionally, Section 4.4 covers further experiments on the scalability of min-product and minmax regret, demonstrating that min-product generally provides better time performance than minmax regret.

4.1. Evaluation metric

We will evaluate the security provided by minmax solution, min-product solution, and the mid-point solution. In addition, we will also include, for comparison, the following defensive strategies: upper-bound, lower-bound, greedy, random, and minmax-product solutions.

The upper-bound and lower-bound solutions represent pessimistic and optimistic defenders. They are similar to the mid-point solution but use the optimization from Section 2.3 on the upper (or lower) bounds of the controls' effectiveness rather than the mid-point of intervals. The greedy solution is a security portfolio that has the largest sum of mid-point controls' effectiveness, subject to budget constraints: Hence, the defender only considers the most effective controls and ignores the placement of controls. The random solution is a random portfolio of controls, subject to budget constraints.

The minmax-product solution is an optimization combining both minmax regret and min-product, and is obtained by adding the product of risk, multiplied by a small value, into the objective function of the minmax regret as a penalty factor. This multi-objective optimization hence considers both the regret and the product of risks when selecting the security portfolio.

Given all possible solutions described so far, we will evaluate their average risk over a set of scenarios. As mentioned, we use the mid-point solution as the benchmark. We denote the average risk of the mid-point solution as r_{mid}, and the relative risk of the other solutions to the mid-point solution is computed using the following equation:

$$\hat{r}_{solution} = (1 - r_{solution}/r_{mid}) \times 100\%. \tag{14}$$

Note that a positive $\hat{r}_{solution}$ indicates the solution outperforms the mid-point solution (i.e., a lower average security risk); otherwise, a negative $\hat{r}_{solution}$ indicates the solution is worse than the mid-point solution (i.e., a higher average security risk).

4.2. Layered random attack graphs

In these experiments, we generate random attack graphs based on a parameter N, the number of nodes in the graph. Each graph includes a source node (node 0) and a sink (node N-1, i.e., the last node). Apart from the source and sink, nodes are distributed in multiple layers, each representing one step in multi-stage attacks. This is inspired by the layers in the MITRE attack matrix, e.g: Reconnaissance, Resource Development, Initial Access, Execution, Persistence, Privilege Escalation, and Defence Evasion, etc. (The MITRE Corporation, 2022).

All nodes within the first layer are connected to the source node, and all nodes in the final layer connect to the sink node. There are two types of edges within the graph: intra-layer and inter-layer. The intra-layer edges are edges within the same layer, representing lateral movement of attackers. We allow each layer to have one intra-layer edge.

The inter-layer edges represent a typical attack step, with a higher probability of 0.3 for edges between nodes in layer i and $i+1$. Finally, each layer has a probability of 0.3 of having one edge with the sink node.

Each layer has its own set of controls, so controls are not repeated along an attack path. Controls are randomly allocated to defend edges in that layer.

Table 2
Average risk and relative risk for the different defence strategies at 50% uncertainty.

Solution	(Nodes, layers)			
	(50,5)	(50,3)	(100,5)	(100,10)
mid-point	0.0710; 0%	0.223; 0.0%	0.154; 0.0%	0.0223; 0.0%
minmax regret	0.0629; 9.9%	0.206; 5.8%	0.139; 5.7%	0.0191; 9.8%
min-product	0.0622; 11.6%	0.206; 5.8%	0.136; 7.0%	0.0191; 11%
minmax-product	0.0628; 9.9%	0.206; 5.8%	0.137; 6.7%	0.0191; 9.8%
lower bound	0.0706; 0.1%	0.223; −0.9%	0.154; −2.4%	0.0237; −15%
upper bound	0.0999; −39.1%	0.245; −10.5%	0.163; −8.9%	0.0257; −18%
greedy	0.418; −557%	0.647; −187%	0.499; −281%	0.112; −1528%
random	0.300; −646%	0.647; −214%	0.657; −393%	0.170; −2811%

Each control has one intensity level with an effectiveness between 0.2 and 0.6. The direct cost of implementing a control is an integer between 1 and 2, and the direct budget is 70% of the total costs. The indirect budget is large. For simplicity, we let the baseline probability be 1 for each edge.

All computations in the experiments were run on a MacBook Pro, with an Apple M2 Pro processor and 16 GB of RAM. The optimization was programmed in Python using the PuLP modeler and Gurobi solver.

4.3. Comparing defence strategies: results

We conduct four sets of experiments with 50% uncertainty in the effectiveness of controls. In each experiment, we generate 20 random attack graphs with different numbers of nodes and layers. For each graph, we randomly generate 500 scenarios: These scenarios are used in the optimization to determine the defensive portfolio. After that, for the evaluation, we generate another 1000 scenarios to compute the average risk returned by each solution across these scenarios. The average risk across 20 graphs is the mean of the average risks for each individual graph. Similarly, the average relative risk of each solution to the mid-point is the mean of the relative risks for each individual graph.

Table 2 shows the results of the experiments: the average risk and average relative risk across all attack graphs. As the table shows, the min-product provides the best security, with an improvement between 5.8% and 11.6% over the baseline, followed by the minmax-product and then the minmax regret. Notice also the dramatic improvement in security when comparing it with "non-optimization based" defences (e.g. mid-point results in an improvement compared to greedy defence of up to 1528%).

4.4. Scalability of min-product and minmax regret

We use random attack graphs with a similar topology to those in Khouzani et al. (2019) and Zhang and Malacaria (2021a) to test the scalability of minmax regret and min-product. These are Erdős-Rényi random attack graphs where the probability that two nodes are connected is $p = 3/|\mathcal{V}|$. The maximal number of edges between two edges is 3. In addition, we modify the random graph to prevent incomplete attack paths that fail to reach the target node. In total, we consider 37 controls, each with two intensity levels. Each edge is associated with one or two controls. The security portfolio can select around 7 controls within the budget.

We conducted two sets of experiments, in the first set using 100 scenarios and in the second set using 500 scenarios. For the 100-scenario experiments, we used 20 graphs with node sizes of 10, 30, 50, 100, 500, 1000. For the 500-scenario experiment, we used 20 graphs with node sizes from 10 to 100 nodes. For 500 nodes, we used 5 graphs.

The running times of the minmax regret are shown in Fig. 3(a) and 3(b). As illustrated, the minmax regret requires a median time of 10 min to find the solution for a random attack graph with 1,000 nodes and 100 scenarios. With 500 nodes and 500 scenarios, the median running time triples.

The running times of the min-product are shown in Fig. 3(c) and 3(d). Min-product is generally more efficient than minmax regret apart from the largest graphs (1000 nodes with 100 scenarios and 500 nodes with 500 scenarios).

5. Multi-dimensional decision support for cybersecurity

In this section we show that the framework we introduced can be used not only to choose controls against a specific threat but also in a more general context of evaluating different cybersecurity investment options and choosing the option of minimal risk.

In this multi-dimensional decision problem, there are n options to choose from, where each option consists of a mixture of products and services. Each option has an associated threat scenario (modeled as an attack graph), where, as usual, for each attack step there are some possible security controls whose effectiveness is given as an uncertainty interval. Also, for each option, there is an associated impact if the corresponding attack is successful.

The objective here is to determine which option and security portfolio to choose with the aim of minimizing the security risk (as usual, subject to budget constraints).

Technically, we model this problem by adding to our framework controls $\gamma_1, \ldots, \gamma_n$ i.e., one control for each option, with the following property: Each new control γ_i is effective on the initial edge incoming to the attack graph associated to option i; each γ_i has cost 0 and has maximal effectiveness, i.e., $\simeq 0$. Cost 0 means that these controls can always be chosen and maximal effectiveness means that if γ_i is chosen, that option is eliminated as the incoming edge to that option is "blocked" by deploying γ_i.

We then add to the optimization the following constraint: $\sum_i x_{\gamma_i} = n - 1$. The role of the above constraint is to force the optimization to select exactly one option: That option is the one for which the minimal objective value is achieved.

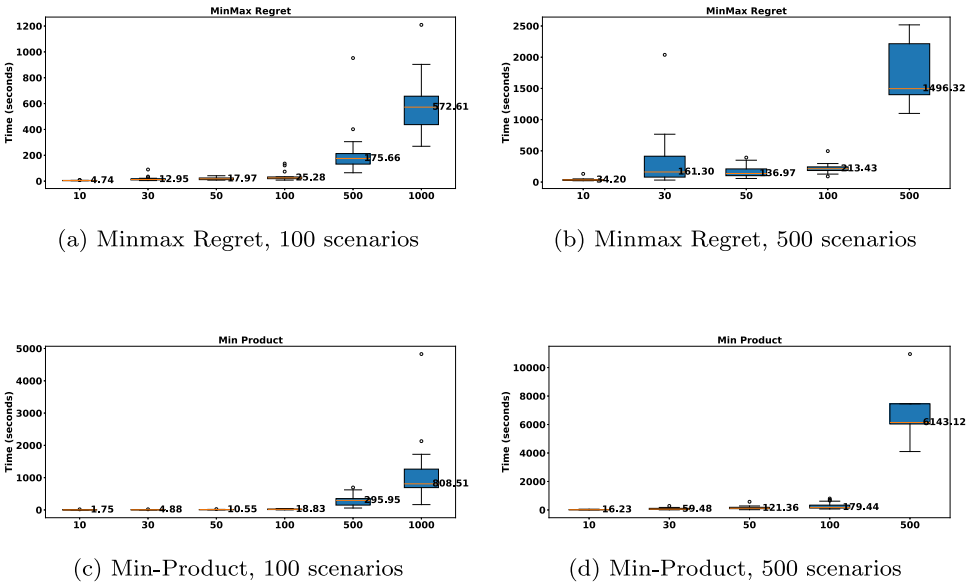
5.0.1. Unsuitability of minmax regret for multi-dimensional modeling

Before moving on to the case study, we ought to observe that there is a problem with using minmax regret for this multi-dimensional modeling. The problem arises from the fact that regret is a ratio between the security of a portfolio in a scenario and the optimal security in that scenario. However, it could happen that the portfolio at the nominator and the one at the denominator select different options: in that case, since the options are mutually exclusive the regret is meaningless and hence the whole minmax regret (and also the minmax regret combined with min-product) result makes no sense. For this reason, in the case study, we will only focus on the min-product and mid-point solutions.

5.1. Case study: IoT home bundle options

Let us consider a decision-making scenario where a home user evaluates three options for a home IoT bundle. The user objective is to choose the security-optimal combination of a home IoT bundle and security controls within budget constraints.

Fig. 4 illustrates the case study with the three home IoT bundle options, each option consists of buying some app-controlled smart LED



(a) Minmax Regret, 100 scenarios (b) Minmax Regret, 500 scenarios

(c) Min-Product, 100 scenarios (d) Min-Product, 500 scenarios

Fig. 3. Scalability results for minmax regret and min-product across 100 and 500 scenarios.

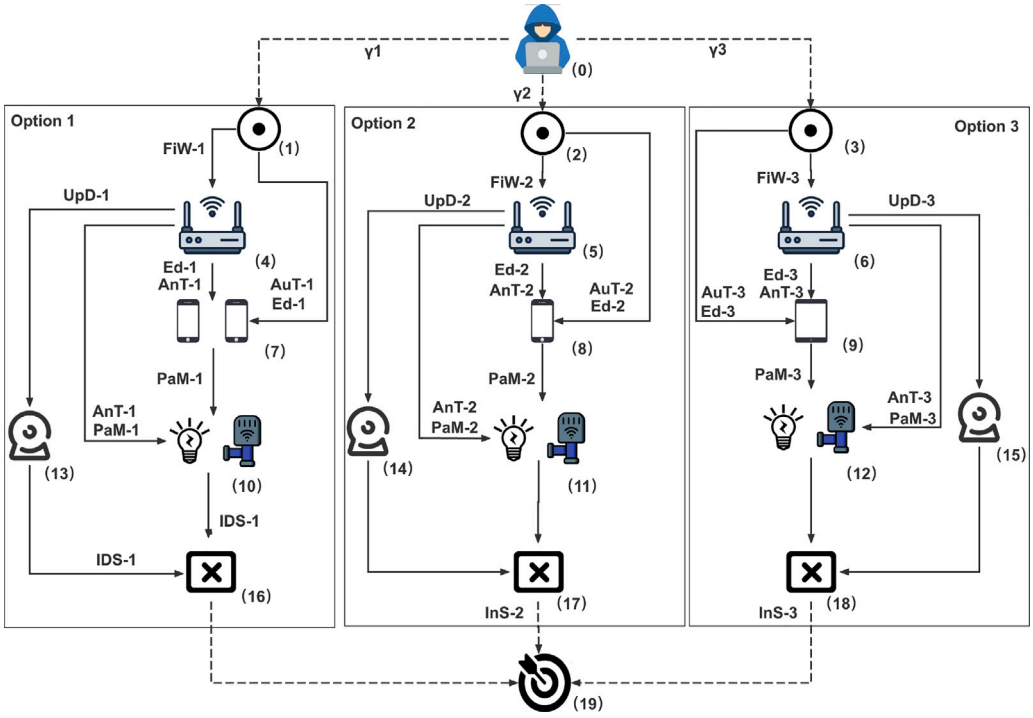


Fig. 4. Case study.

and smart thermostats. Each option also includes a legacy device (an old security camera) which is more vulnerable than the LED and thermostat. Each bundle option shares a similar attack graph topology: In each of the three cases, the attacker can begin the attack by exploiting vulnerabilities in the home router to gain a foothold in the network. Next, the attacker can exploit the apps to access the smart home devices or directly exploit the smart home devices from the router. Alternatively, the attacker can launch social engineering attacks to get valid credentials for the apps and so access the smart home devices. The attacker could also choose to attack the more vulnerable legacy device.

Let us take a closer look at each possible option:

Option 1 (left): In this option, the home user buys LEDs and thermostats, which are controlled by different apps (e.g., they may come from different companies using different standards). In addition, in this option, the user does not buy a cybersecurity insurance plan to compensate for losses in case of a security breach. This option however includes the possibility of buying a very effective, yet expensive, custom intrusion detection system (IDS).

Option 2 (middle): The attacker graph structure is similar to option 1, but the smart home devices are here controllable within one integrated app. In addition, option 2 offers security insurance to hedge the risk of a security breach. However, the IDS is not provided.

Table 3
Range of control effectiveness.

Control	Option 1	Option 2	Option 3
Ed	L - H	L - H	M - VH
FiW	L - H	L - H	L - H
PaM	M - VH	L - H	L - M
AnT	L - H	L - H	L - H
AuT-L1	M - H	M - H	M - H
AuT-L2	H - VH	H - VH	H - VH
UpD	VL - VH	VL - VH	VL - VH
IDS	H - VH	n/a	n/a
Ins	n/a	H	H

Option 3 (right): This is similar to the middle option and offers users security insurance. However, in this option, every family member (children included) is allowed to have access to the app controlling the smart home devices. In addition, the device can store more sensitive users' data.

In addition, allowing every family member to have access to the app makes option 3 more vulnerable to social engineering attacks. Furthermore, security patch management can be challenging for option 1; for example, updates for one app may be missed while another has been updated. However, in option 1, the IoT devices have strong patch management support.

Security Controls: there are multiple security controls that can mitigate the security risk. We use five levels indicating the security control effectiveness of reducing risk $VL = 0.9$, $L = 0.7$, $M = 0.5$, $H = 0.3$, $VH = 0.1$ (the lower the value, the higher the effectiveness).

- **User Education (Ed)**: security awareness training and education.
- **Firewall (FiW)**: a firewall protects all devices on a home network by blocking harmful traffic.
- **Patch Management (PaM)**: regularly managed updates and patches for the apps.
- **Antivirus Software (AnT)**: a software that detects, prevent and remove malware.
- **Authentication (AuT)**:
  - Level 1: passwords for the apps.
  - Level 2: two-factor authentication (2FA) for the apps.
- **Intrusion Detection System (IDS)**: a security tool to detect anomalies within the network.
- **Firmware Update (UpD)**: updates firmware regularly if possible.
- **Cybersecurity Insurance (Ins)**: a cybersecurity plan to compensate for losses in case of a security breach.

Table 3 shows the range of the controls' effectiveness for each option; these may differ based on the product and configuration.

The baseline probability of edge $3 \rightarrow 9$ is VL, since option 3, which allows all family members to use the app, is more vulnerable to social engineering attacks. The baseline probability of edges $1 \rightarrow 7$, $2 \rightarrow 8$, $4 \rightarrow 13$, $5 \rightarrow 14$, $6 \rightarrow 15$ is L, i.e., Social engineering attacks with a large number of attempts make users vulnerable, and the out-dated security camera has little security support from the manufacturer. The baseline probabilities of edges $4 \rightarrow 7$, $5 \rightarrow 8$, and $6 \rightarrow 9$ are H, with secure communication between the router and the apps. The baseline probability of $0 \rightarrow 1$, $0 \rightarrow 2$, $0 \rightarrow 3$, $13 \rightarrow 16$, and $14 \rightarrow 17$, $15 \rightarrow 18$ are set to 1, as they are "auxiliary" edges that do not represent an actual attack step. The remaining edges have a baseline probability of M.

We categorize the impact in two levels: $HI = 1$ and $LI = 0.5$, i.e., the greater the value, the greater the impact. The impact of a security breach for option 3 is considered high (HI), as the devices may store more sensitive users data, resulting in a more significant impact if compromised. The remaining two options are low LI. Applying control **Ins** can hedge the impact of a security breach.

We assume the direct budget is sufficiently large. The indirect cost of security controls is as follows: each control costs one budget, except **2FA** (level 2 authentication) which costs two. In addition, control **IDS** in option 1 costs four.

In the experiments, we compare the min-product and mid-point portfolios subject to the indirect budget in a range from 2 to 8. As mentioned, we exclude from the comparison minmax regret and minmax regret with product because the nominator and denominator of regret may be in different options. We randomly generate 500 scenarios to determine the portfolios and test the solutions on a different set of 1000 scenarios (both sets of scenarios use the same uncertainty intervals as from Table 3). The solutions of mid-point and min-product portfolios for budgets in the range [2,8] are presented in Table 4: As the table shows, option 2 is the better option for lower budgets, with appropriate controls available to mitigate the risks on all attack paths. At higher budgets, i.e. when control IDS can be purchased, option 1 becomes the better option.

6. Conclusions and further works

This paper explored the resilience of cybersecurity decision-making models to uncertainties in security controls' effectiveness. We introduced and compared minmax regret and min-product of risks, finding that the latter offers better resilience. A case study on home IoT cybersecurity investments demonstrated the practical utility of our approach.

Integrating real-time data with machine learning to refine security control metrics, applying the framework to different domains, and considering human factors in decision-making processes are potential areas for further study. Developing user-friendly tools for practitioners would also enhance practical adoption.

CRediT authorship contribution statement

Yunxiao Zhang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Pasquale Malacaria:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used Grammarly and ChatGPT in order to improve language and readability, with caution. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the EPSRC grants (EP/T026596/1) under the "CHAI: Cyber Hygiene in AI enabled domestic life" project.

Data availability

No data was used for the research described in the article.

Table 4
Solutions at budgets 2,...,8.

Budget	Solution	Portfolio	Avg. Risk
2	mid-point	Option 2: Ins, UpD	0.0314 (0%)
2	min-product	Option 2: FiW, Ins	0.0287 (8.72%)
3	mid-point	Option 2: Ins, UpD	0.0316 (0%)
3	min-product	Option 2: PaM, Ins, UpD	0.0271 (13.98%)
4	mid-point	Option 2: Ed, FiW, Ins, UpD	0.0165 (0%)
4	min-product	Option 2: FiW, AuT, Ins, UpD	0.0153 (7.22%)
5	mid-point	Option 2: FiW, AuT, Ins, UpD	0.0151 (0%)
5	min-product	Option 2: FiW, PaM, AuT, Ins, UpD	0.0135 (10.62%)
6	mid-point	Option 2: FiW, AuT, Ins, UpD	0.0150 (0%)
6	min-product	Option 2: FiW, PaM, AuT-L2, Ins, UpD	0.0133 (11.34%)
7	mid-point	Option 1: Ed, FiW, UpD, IDS	0.0108 (0%)
7	min-product	Option 1: FiW, PaM, UpD, IDS	0.00938 (13.14%)
8	mid-point	Option 1: Ed, FiW, UpD, IDS	0.0109 (0%)
8	min-product	Option 1: FiW, PaM, AuT, UpD, IDS	0.00891 (18.84%)

References

Abdallah, M., Woods, D., Naghizadeh, P., Khalil, I., Cason, T., Sundaram, S., Bagchi, S., 2021. Morshed: Guiding behavioral decision-makers towards better security investment in interdependent systems. In: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. pp. 378–392.

Ahmad, I., Jayswal, A., Banerjee, J., 2013. On interval-valued optimization problems with generalized invex functions. J. Inequal. Appl. 2013 (1), 1–14.

Banga, G., 2020. Why is cybersecurity not a human-scale problem anymore? Commun. ACM 63 (4), 30–34.

Ben-Tal, A., El Ghaoui, L., Nemirovski, A., 2009. Robust Optimization, vol. 28, Princeton University Press.

Bhuiyan, T.H., Nandi, A.K., Medal, H., Halappanavar, M., 2016. Minimizing expected maximum risk from cyber-attacks with probabilistic attack success. In: 2016 IEEE Symposium on Technologies for Homeland Security. HST, IEEE, pp. 1–6.

Cavusoglu, H., Raghunathan, S., Yue, W.T., 2008. Decision-theoretic and game-theoretic approaches to IT security investment. J. Manage. Inf. Syst. 25 (2), 281–304.

Charnes, A., Granot, F., Phillips, F., 1977. An algorithm for solving interval linear programming problems. Oper. Res. 25 (4), 688–695.

Chronopoulos, M., Panaousis, E., Grossklags, J., 2017. An options approach to cybersecurity investment. IEEE Access 6, 12175–12186.

Durkota, K., Lisỳ, V., Kiekintveld, C., Bošanskỳ, B., 2015. Game-theoretic algorithms for optimal network security hardening using attack graphs. Database 20, 4xPC.

Fang, F., Nguyen, T., Pickles, R., Lam, W., Clements, G., An, B., Singh, A., Tambe, M., Lemieux, A., 2016. Deploying paws: Field optimization of the protection assistant for wildlife security. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, No. 2. pp. 3966–3973.

Fielder, A., König, S., Panaousis, E., Schauer, S., Rass, S., 2018. Risk assessment uncertainties in cybersecurity investments. Games 9 (2), 34.

Fielder, A., Panaousis, E., Malacaria, P., Hankin, C., Smeraldi, F., 2016. Decision support approaches for cyber security investment. Decis. Support Syst. 86, 13–23.

Gordon, L.A., Loeb, M.P., 2002. The economics of information security investment. ACM Trans. Inf. Syst. Secur. 5 (4), 438–457.

Kasperski, A., Zieliński, P., 2006. An approximation algorithm for interval data minmax regret combinatorial optimization problems. Inform. Process. Lett. 97 (5), 177–180.

Khouzani, M., Liu, Z., Malacaria, P., 2019. Scalable min-max multi-objective cybersecurity optimisation over probabilistic attack graphs. European J. Oper. Res. 278 (3), 894–903.

Khouzani, M., Malacaria, P., Hankin, C., Fielder, A., Smeraldi, F., 2016. Efficient numerical frameworks for multi-objective cyber security planning. In: Computer Security–ESORICS 2016: 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part II 21. Springer, pp. 179–197.

Kiekintveld, C., Islam, T., Kreinovich, V., 2013. Security games with interval uncertainty. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems. AAMAS '13, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 231–238.

Korzhyk, D., Yin, Z., Kiekintveld, C., Conitzer, V., Tambe, M., 2011. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. J. Artificial Intelligence Res. 41, 297–327.

Kouvelis, P., Yu, G., 2013. Robust Discrete Optimization and Its Applications, vol. 14, Springer Science & Business Media.

La, Q.D., Quek, T.Q., Lee, J., Jin, S., Zhu, H., 2016. Deceptive attack and defense game in honeypot-enabled networks for the internet of things. IEEE Internet Things J. 3 (6), 1025–1035.

Letchford, J., Vorobeychik, Y., 2013. Optimal interdiction of attack plans. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems. AAMAS '13, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 199–206.

Nandi, A.K., Medal, H.R., Vadlamani, S., 2016. Interdicting attack graphs to protect organizations from cyber attacks: A bi-level defender–attacker model. Comput. Oper. Res. 75, 118–131.

Paruchuri, P., Pearce, J.P., Marecki, J., Tambe, M., Ordonez, F., Kraus, S., 2008. Playing games for security: An efficient exact algorithm for solving Bayesian stackelberg games. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2. pp. 895–902.

Pendleton, M., Garcia-Lebron, R., Cho, J.-H., Xu, S., 2016. A survey on systems security metrics. ACM Comput. Surv. 49 (4), 1–35.

Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., Kraus, S., 2008. Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track. pp. 125–132.

Pita, J., Jain, M., Ordóñez, F., Tambe, M., Kraus, S., Magori-Cohen, R., 2009. Effective solutions for real-world stackelberg games: When agents must deal with human uncertainties. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. pp. 369–376.

Pita, J., John, R., Maheswaran, R., Tambe, M., Kraus, S., 2012. A robust approach to addressing human adversaries in security games. In: ECAI 2012. IOS Press, pp. 660–665.

Rass, S., König, S., Schauer, S., 2015. Uncertainty in games: Using probability-distributions as payoffs. In: Decision and Game Theory for Security: 6th International Conference, GameSec 2015, London, UK, November 4-5, 2015, Proceedings 6. Springer, pp. 346–357.

Sawik, T., 2013. Selection of optimal countermeasure portfolio in IT security planning. Decis. Support Syst. 55 (1), 156–164.

Sawik, T., 2022. A linear model for optimal cybersecurity investment in Industry 4.0 supply chains. Int. J. Prod. Res. 60 (4), 1368–1385.

Scott, E., Panda, S., Loukas, G., Panaousis, E., 2022. Optimising user security recommendations for AI-powered smart-homes. In: 2022 IEEE Conference on Dependable and Secure Computing. DSC, IEEE, pp. 1–8.

Sinha, A., Malo, P., Deb, K., 2017. A review on bilevel optimization: From classical to evolutionary approaches and applications. IEEE Trans. Evol. Comput. 22 (2), 276–295.

Smeraldi, F., Malacaria, P., 2014. How to spend it: optimal investment for cyber security. In: Proceedings of the 1st International Workshop on Agents and CyberSecurity. pp. 1–4.

Smith, J.C., Song, Y., 2020. A survey of network interdiction models and algorithms. European J. Oper. Res. 283 (3), 797–811.

Steuer, R.E., 1981. Algorithms for linear programming problems with interval objective function coefficients. Math. Oper. Res. 6 (3), 333–348.

The MITRE Corporation, 2022. MITRE att&ck matrix for enterprise. https://attack.mitre.org/matrices/enterprise/.

The National Cyber Security Centre, 2023. The NCSC research problem book. https://www.ncsc.gov.uk/collection/problem-book.

Tsiodra, M., Panda, S., Chronopoulos, M., Panaousis, E., 2023. Cyber risk assessment and optimisation: A small business case study. IEEE Access 11, 44467–44481.

Uuganbayar, G., Yautsiukhin, A., Martinelli, F., Massacci, F., 2021. Optimisation of cyber insurance coverage with selection of cost effective security controls. Comput. Secur. 101, 102121.

Verendel, V., 2009. Quantified security is a weak hypothesis: A critical survey of results and assumptions. In: Proceedings of the 2009 Workshop on New Security Paradigms Workshop. NSPW '09, Association for Computing Machinery, New York, NY, USA, pp. 37–50.

Wu, H.-C., 2008. On interval-valued nonlinear programming problems. J. Math. Anal. Appl. 338 (1), 299–316.

Yang, R., Kiekintveld, C., Ordonez, F., Tambe, M., John, R., 2011. Improving resource allocation strategy against human adversaries in security games. In: IJCAI Proceedings-International Joint Conference on Artificial Intelligence, Vol. 22, No. 1. Barcelona, p. 458.

Yin, Z., Tambe, M., 2012. A unified method for handling discrete and continuous uncertainty in bayesian stackelberg games. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2. Citeseer, pp. 855–862.

Zhang, Y., Malacaria, P., 2021a. Bayesian Stackelberg games for cyber-security decision support. Decis. Support Syst. 148, 113599.

Zhang, Y., Malacaria, P., 2021b. Optimization-time analysis for cybersecurity. IEEE Trans. Dependable Secure Comput. 19 (4), 2365–2383.

Zhang, Y., Malacaria, P., 2023. Keep spending: Beyond optimal cyber-security investment. In: 2023 IEEE 36th Computer Security Foundations Symposium. CSF, IEEE, pp. 123–136.

Zhang, Y., Malacaria, P., Loukas, G., Panaousis, E., 2023. CROSS: a framework for Cyber Risk Optimisation in Smart homes. Comput. Secur. 130, 103250.

Żychowski, A., Mańdziuk, J., 2021a. Evolution of strategies in sequential security games. In: Proceedings of the 20th AAMAS Conference. pp. 1434–1442.

Żychowski, A., Mańdziuk, J., 2021b. Learning attacker's bounded rationality model in security games. In: International Conference on Neural Information Processing. Springer, pp. 530–539.

Yunxiao Zhang received the B.Eng. degree in Electrical and Electronic Engineering from Newcastle University, UK, and the M.Sc. degree in Control Systems and the Ph.D. degree from Imperial College London, UK. He was a postdoctoral researcher at Queen Mary University of London. Currently, he is a Lecturer in Cyber Security at the Department of Computer Science, University of Exeter, UK. His current research interests lie in the economics of cybersecurity, optimisation, and game theory.

Pasquale Malacaria received his Laurea in Philosophy from "La Sapienza" University in Rome and his Ph.D. from the University of Paris VII in France. His work focuses on information theory, optimization, game theory, verification and their applications to computer security. He is a Professor of Computer Science at Queen Mary University of London. He has been an EPSRC advanced research fellow, is a recipient of the Alonzo Church award 2017 and the Facebook Faculty awards 2015.