Computers & Security 130 (2023) 103250

Contents lists available at ScienceDirect

Computers & Security

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

CROSS: A framework for cyber risk optimisation in smart homes

Yunxiao Zhang[a,*], Pasquale Malacaria[a], George Loukas[b], Emmanouil Panaousis[b]

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

ARTICLE INFO

Article history:
Received 1 November 2022
Revised 29 March 2023
Accepted 3 April 2023
Available online 5 April 2023

Keywords:
Smart home security
Mathematical optimisation
Security controls
IoT
Artificial intelligence

ABSTRACT

This work introduces a decision support framework, called Cyber Risk Optimiser for Smart homeS (CROSS), which advises both smart home users and smart home service providers on how to select an optimal portfolio of cyber security controls to counteract cyber attacks in a smart home including traditional cyber attacks and adversarial machine learning attacks. CROSS is based on a multi-objective bi-level two-stage optimisation. In stage-one optimisation, the problem is modelled as a multi-leader-follower game that considers both security and economic objectives, where the provider selects a security portfolio to protect both itself and its users, while rational attackers target the weakest path. Stage-two optimisation is a Stackelberg security game that focuses on additional user security controls under the remit of smart home users. While CROSS can potentially be applied to other similar use cases, in this paper, our aim is to address threats against artificial intelligence (AI) applications as the use of AI in smart Internet of Things (IoT) devices introduces new cyber threats to home environments. Specifically, we have implemented and assessed CROSS in a smart heating use case in a prototypical AI-enabled IoT environment that combines characteristics and vulnerabilities currently present on existing commercial off-the-shelf (COTS) devices, demonstrating the selection of optimal decisions.

1. Introduction

Devices such as smart speakers, smart thermostats, security cameras with face recognition, and in the near future, brain-computer interfaces and elderly care companion robots are all empowered by AI algorithms to offer competitive advantages over legacy technologies. AI can bring considerable benefits, including energy efficiency, comfort, and even health. While we all enjoy these benefits in our domestic lives, these technologies bear significant risks and AI capabilities adversely affect our smart homes by increasing the attack surface (Bispham et al., 2019; Yan et al., 2022). In light of this, any provider of an AI service faces the challenge of determining optimal security controls. Given the proliferation of attacks against machine learning, the scope of this decision about security controls is enlarged to include controls against adversarial machine learning (AdvML).

In this paper, we propose an original decision support system, called Cyber Risk Optimiser for Smart homeS (CROSS) aimed both at the users and the smart home providers. CROSS selects an optimal portfolio of traditional cybersecurity and AdvML controls, henceforth referred to as security controls, to counteract multi-stage attacks in the smart home supply chain.

CROSS considers the whole defence problem as a two-stage optimisation problem[1]. In stage one, the provider selects an optimal security portfolio to minimize risk and maximize profit, while users decide whether to accept the service. In stage two, the provider recommends optimal user controls for extra security, which may require user commitment. Users decide whether to commit based on their preferences for quality of experience (QoE) and security.

To the best of our knowledge, no prior work has focused on studying the optimal security control problem for smart home users, which involves both the users and the service provider. This problem is significant because smart home devices, in particular those with AI capabilities, introduce new cyber security threats to users who cannot plan an optimal defence strategy alone. Thus, we

* Corresponding author.
E-mail addresses: yunxiao.zhang@qmul.ac.uk (Y. Zhang), p.malacaria@qmul.ac.uk (P. Malacaria), g.loukas@greenwich.ac.uk (G. Loukas), e.panaousis@greenwich.ac.uk (E. Panaousis).

[1] Notice there are three optimisations in CROSS, because stage one also includes an optimisation to determine maximal profit constraints, which are then used in the security optimisation for the service provider.

https://doi.org/10.1016/j.cose.2023.103250

consider the service provider who plays a significant role in cybersecurity defence in the smart home environment.

The main contributions and challenges to our work are as follows:

- We solve the provider's defence problem as a two-stage optimisation:
  - Stage-one optimisation is modelled as a multi-leader-follower game that considers both the security risks in a smart home, and the smart home user preferences. In addition, we provide the maximal profit constraints to guarantee the provider's target profit, i.e. the provider is willing to provide the service only if the expected profit from users is not less than the target profit.
  - Stage-two optimisation is formulated as a Stackelberg security game, where the service provider suggests an optimal portfolio of user controls as an additional layer of security for smart home users based on their preferences. These controls must be implemented by the users themselves and not by the service provider.
- The two-stage optimisation problem is a non-linear bi-level optimisation problem which is NP-hard. To solve such a problem, we use the property of totally unimodular matrices and strong duality to convert the non-linear bi-level optimisation into a tractable MICP for both stage-one optimisation and stage-two optimisation. Thanks to the recent advances in mixed-integer optimisation (Lubin, 2017; Morán R et al., 2012), we can efficiently solve a MICP using existing solvers (e.g. MOSEK version 9.2).
- There is a lack of case studies on control optimisation for smart home security, including protecting devices that utilise AI technologies. Thus, we create a realistic use case study of an AI-powered smart radiator valve along with an exhaustive attack graph consisting of all the required parameters (attack techniques and security controls). We use such a use case study to assess the performance of the proposed decision-support system CROSS.

The paper is structured as follows: we review related work in Section 2, covering topics such as security games, security investment and supply chain, and adversarial machine learning attacks. Section 3 introduces the system model, including attack graphs and security portfolios. We also provide a list of symbols in Table 1. In Section 3.3, we present the two-stage defence problem, along with an example in Section 3.4. We then analyse the optimality of the two-stage defence problem and present efficient solutions in Section 4. To help readers better understand the two-stage defence problem, we provide a case study in Section 5, which models a prototypical AI-enabled Internet of Things (IoT) environment. Finally, we conclude our paper in Section 6.

2. Related work

2.1. Security games

Stackelberg game-theoretical approaches have been widely applied for solving the security challenges, such as public safety (e.g. ARMOR (Pita et al., 2008) and PROTECT (Shieh et al., 2012)), wildlife protection (e.g. PAWS (Fang et al., 2016)), and particularly cybersecurity defence (Durkota et al., 2015; Fielder et al., 2016; Khouzani et al., 2019; 2016; Sawik, 2013; Vaněk et al., 2012; Zhang and Malacaria, 2021; 2022). In such games, the defender acts first as a leader who anticipates the best response of attackers and therefore commits to an optimal defence strategy that maximises the defender's payoff. Next, the followers (attackers of potentially different types) observe the implemented defence strategy and find the optimal attacking strategy to maximise their payoffs accordingly. Moreover, in these security challenges, the defender always has limited security resources or budgets to protect all possible targets or vulnerabilities at all times. In other words, the defender also wants to minimise the expenditure on an optimal defence strategy. Thus these games can be formulated as a multi-objective bi-level optimisation problem. One highly relevant work to this paper is Khouzani et al. (2019), where the cybersecurity defence problem is modelled as a multi-objective bi-level Stackelberg game with a probabilistic attack graph.

Further work in Zhang and Malacaria (2022) provides a mathematical framework including both a Markov chain and attack graphs to reason about time resilience for cybersecurity and optimal defence. The models in Fielder et al. (2016); Khouzani et al. (2016) have similar game-theoretical approaches, but they only consider single-step attacks rather than multi-step attack scenarios. Nevertheless, in all these papers, the authors have not investigated the smart home supply chain problem that we address in this paper. Moreover, they have not studied multiple attacker types as well as multiple users to be protected by the recommended optimal security portfolio. Other research that uses such security games to tackle cybersecurity challenges includes: Vaněk et al. (2012) that studies the optimal resource allocation problem for packet selection and inspection to detect potential cyber-threats; Zonouz et al. (2013) that introduces a game-theoretic automatic intrusion response engine; Cavusoglu et al. (2008) that proposes another game-theoretic approach for determining IT security investment levels, and Durkota et al. (2015) that uses a security game approach to find the optimal number of honeypots to be placed in computer networks.

When a defender needs to face multiple attacker types, the security games belong to a special class, known as *Bayesian Stackelberg games*. Such a game may arise in a cybersecurity challenge because, for example, the defender has uncertain knowledge about the attacker types who possess varying capabilities to compromise the defender. Several solutions have been proposed to solve a general Bayesian Stackelberg game efficiently (Jain et al., 2011; Paruchuri et al., 2008; Yin and Tambe, 2012). More importantly, a recent study (Zhang and Malacaria, 2021) applies the properties of totally unimodular matrices and strong duality to convert a cybersecurity game (Bayesian Stackelberg game) into a tractable MICP. Such a scalable approach can efficiently solve a cybersecurity game over large attacker graphs to find optimal security portfolios. A more detailed review of game theory applied to security challenges is in Do et al. (2017).

Additionally, in Sanjab et al. (2017), a network interdiction game between a vendor and attacker in a drone delivery system was analysed, incorporating Prospect Theory to capture subjective decision-making. However, the *network interdiction* is commonly used to analyse security games where the goal is to disrupt or prevent the flow of resources through a network. In most cybersecurity problems it is not realistic to completely interdict an attack path. Our approach focuses hence on mitigating security risks by implementing security controls to reduce the success probability of attackers.

2.2. Security investment and supply chain

The general problem of cybersecurity investment has been studied in a handful of papers. One of the initial works is in Gordon and Loeb (2002) that the model considers both the costs and benefits in the optimal security investment. As mentioned in the previous section, the authors in Cavusoglu et al. (2008) propose a game-theoretical approach for a firm determining IT security investment levels and compare it to a decision theory approach. Fur-

ther work (Sawik, 2013) uses financial engineering tools in IT security planning. Since then, there have been a number of studies (Chronopoulos et al., 2017; Fielder et al., 2016; Khouzani et al., 2019; 2016; Panda et al., 2020; Zhang and Malacaria, 2021; 2022) to address both the theoretical and practical cybersecurity investment problems. In addition, Abdallah et al. (2021) investigates the behavioural biases of human decision-making in security investment problems.

The latest studies (Li and Xu, 2020; Sawik, 2020; 2021) are now focusing on security investment for supply chains: Sawik (2020) linearises the classic exponential function of the breach probability to select optimal safeguards for Industry 4.0 supply chains; Sawik (2021) develops an efficient solution to simultaneously mitigate both the direct and indirect propagated risks in a multi-tier supply chain; and Li and Xu (2020) studies the risk propagation problem with a two-echelon supply chain.

2.3. Adversarial machine learning (AdvML) attacks

Machine learning (ML) techniques have had huge success in a wide range of applications we use today. The growing number of vulnerabilities in ML introduces significant cybersecurity concerns, which have been studied by a number of the AdvML literature, e.g. Biggio and Roli (2018); Carlini et al. (2016); Chakraborty et al. (2021); The MITRE Corporation. MITRE ATLAS (2021); Gu et al. (2017); Liu et al. (2018); Pitropakis et al. (2019); Szegedy et al. (2013); Tabassi et al. (2019); Zhang et al. (2017). In Szegedy et al. (2013), the authors first let well-performed neural networks misclassify an image by applying a perturbation. Later, researchers also discovered vulnerabilities in automatic speech recognition and voice controllable systems (Carlini et al., 2016; Zhang et al., 2017). Work in Gu et al. (2017) showed how to let a sign classifier identify stop signs as speed limits by physically adding a crafted sticker to the stop sign. Since these studies have drawn a lot of attention, there has been a significant number of surveys on security evaluation to AdvML (Biggio and Roli, 2018; Liu et al., 2018; Pitropakis et al., 2019) as well as defensive techniques (Chakraborty et al., 2021; Tabassi et al., 2019). Most recently, the MITRE association has published a knowledge base of adversarial tactics, techniques, and real-world case studies for security groups and academic researchers, called MITRE ATLAS (The MITRE Corporation. MITRE ATLAS, 2021). These bespoke, to AI systems, attacks are complementary to those in the MITRE ATT&CK framework (The MITRE Corporation, 2022).

The AdvML techniques can be characterised with respect to the system operation stages (Tabassi et al., 2019). These attack techniques applied to the system's training stage will attempt to adversely change or acquire the training data or the ML model itself. Further, instead of tampering with the training data or the model, the attack techniques applied to the system testing/inference stage focus on generating adversarial examples as inputs to evade the classification. The consequences of a successful AdvML technique can be characterised as the violations of integrity (data misclassification or confidence reduction), availability (unacceptable speed or unusable outputs), confidentiality (stolen model or data), and privacy (stolen personal identifiable information) (Biggio and Roli, 2018; Tabassi et al., 2019).

Similarly, the defence techniques, against AdvML, are also characterised with respect to the system operation stages (Chakraborty et al., 2021; Tabassi et al., 2019). In the training stages, the defences against poisoning attacks include *Data Sanitisation* which removes examples causing high error rates, and *Robust Statistics* which uses constraints and regularisation to reduce potential distortion of the model. In the testing/inference stages, several techniques improve the model robustness, such as *Adversarial Training, Gradient Masking, Defensive Distillation, Ensemble Methods, Feature Squeezing*, and *Reformers/Autoencoders*. In addition, it also includes randomisation mechanisms, such as *Differential Privacy*. However, these defence techniques often cause performance overhead and impacts on model accuracy (Chakraborty et al., 2021). We use direct and indirect costs to evaluate these overheads and impacts for the defender choosing the security portfolio. Moreover, the MITRE ATLAS summarises possible traditional controls (e.g. staff training, antimalware, etc.) that can also help mitigate the AdvML attacks.

3. System model

The CROSS model consists of three types of players: (i) the provider of an AI service or product (we use these terms interchangeably), available to smart home users. The provider (who acts as the defender) seeks to minimise its product's security risks while economically maximising its profit. (ii) smart home users – the potential clients of the provider; and (iii) an attacker representing possible threat actors exploiting the product to achieve adversarial goals. We also refer to this player as the attackers due to the different attacker types studied. We propose a novel method to support the provider as well as each user with optimal decisions necessary in the presence of a cyber threat.

Fig. 1 illustrates the kind of threat scenario this paper addresses. Node S represents the source, i.e. the initial state before any attack is launched. Nodes C_1 to C_K are K possible consequences when the attacker has successfully exploited the service, e.g. violations of integrity, availability, confidentiality, or privacy (Tabassi et al., 2019). Each will cause the corresponding loss to the provider. Node T is an auxiliary sink node and does not represent any privileged state that could be compromised by the attacker. The consequence nodes lead to this auxiliary sink node, allowing us to have a complete path from the Source to the Target in the attack graph.

In addition, we consider N attacker types and, for each attacker type i, we denote the probability of this type occurring by a_i. This probability can be drawn from either some threat intelligence or publicly available reports on cyber attacks against smart homes. For example, the *NCSC*'s Cyber Security Breaches Survey 2022 (The National Cyber Security Centre, 2022) provides general data on the types of cyber-attacks faced by UK organisations, as well as their impact and responses. Moreover, the consumer protection group *Which* reported, using a honeypot-based experiment, that a smart home could be exposed to more than 12,000 hacking or unknown scanning attacks in a week (Which, 2021). In the first half of 2021, cybersecurity company *Kaspersky* detected 1.5 billion IoT attacks, partly attributing an increase over the previous year to the proliferation of smart home devices (PYMNTS, 2021). However, these works may not be detailed enough for the service providers in the proposed decision support system. The service providers should conduct their surveys to gain a more accurate understanding of the probability of attack types against smart homes.

In Fig. 1, we use rectangles to represent attack graphs, each graph consisting of attack paths the attacker can traverse toward exploiting the assets of the provider or the user. In Section 3.1, we will describe these attack graphs. An example of an attack graph is also illustrated in Section 3.4. Although different attacker types share the same attack graph, each type i has a specific baseline success probability to exploit underlying vulnerabilities when there is no defensive control to protect the product as a result of its unique attacking capabilities. For example, the attacker could be an expert in launching attacks against the AI applications of the smart home services, thus having a higher success rate for those exploitation actions (i.e. edges in the attack graph) that represent AdvML

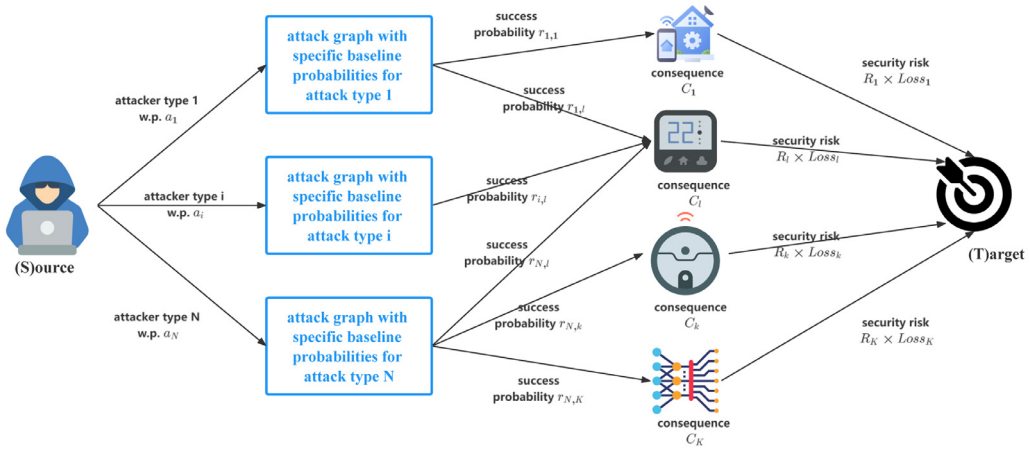

Fig. 1. Modelling threat scenario using attack graphs.

techniques. The highest probability that attacker type i can successfully perform a whole sequence of exploitation actions, which forms a complete attack path leading to consequence node C_k, is denoted by $r_{i,k}$. A formal definition of $r_{i,k}$ is later given in (19). Notice that a successful attacker type can lead to multiple consequences and one consequence can be caused by more than one attacker type.

As a result, a consequence may cause a loss to the provider. Here, we denote R_k as the highest success probability that the attacker can cause consequence k, i.e. $R_k = \max_i\{r_{i,k}\}$. The security risk associated with consequence C_k is defined as the product of R_k and the impacts Loss_k (refereed as the *direct loss*), defined in Eq. (5). Such attacks may propagate to users, where the user's loss is defined in Eq. (7).

We present a list of symbols in Table 1.

3.1. Attack graphs

We use probabilistic attack graphs to model the security risk of the provider and the users. Nodes in an attack graph represent the privilege states of the attacker, and every edge represents an exploitation action which allows the attacker to escalate his privilege state. Each edge is associated with an exploitation risk (e.g. due to underlying vulnerabilities or lack of security controls), and the success rate of an exploitation action is determined by both the effectiveness of applied security controls and the underlying vulnerabilities. In a privilege state, the attacker can select from one of the next exploitation actions corresponding to directed edges leaving the state.

Such a probabilistic attack graph is defined as a directed multigraph $G = (\mathcal{V}, \mathcal{E}, \overline{h}, \underline{t}, p, s, \mathcal{T})$: $\mathcal{V}$ denotes the set of nodes; $\mathcal{E}$ denotes

Table 1
List of symbols.

Symbol	Description
	Functions and Sets
$\mathcal{A}_k$	set of attacker types that can cause consequence k;
$\mathcal{C}$	set of all security controls;
$\mathcal{C}(e)$	subset of controls that are effective on edge e;
$\mathcal{E}$	set of edges;
$\mathcal{L}(c)$	the set of intensity levels of control c;
$\mathcal{T}$	set of target nodes
$\mathcal{V}$	set of nodes;
$\overline{h}, \underline{t}$	functions that returns "head" and "tail" of an edge;
	Constants
a_i	occurrence probability of attacker type i;
B_D, B_{ID}	direct and indirect cost budgets for the provider;
$\texttt{Cost}_{cl}$	direct cost of control c at level l for the provider,;
$\texttt{IndirectCost}_{cl}$	indirect cost of control c at level l for the provider,;
$\texttt{IndCost}_{clm}$	indirect cost for the user type m of control c at level l;
K	total number of consequences;
$\texttt{Loss}_k$	direct loss of the provider if consequence k happens;
N	total number of attacker types;
M	total number of user types;
l	control level l;
$\pi_e(i)$	base-line probability of edge e being exploited by attacker type i;
p_{ecl}	effectiveness of control c at level l on edge e;
p_m	premium paid (per user) by user type m;
P_m	total profit gained from user type m;
$\overline{P}$	target profit of the provider;
U_m	the number of user type m who sign up with the offered service;
$u_{I,m}$	user m tolerance thresholds for indirect costs;
$u_{UL,m}$	user m tolerance thresholds for security risks;
$\texttt{userLoss}_{km}$	loss of user type m if consequence k happens;

(*continued on next page*)

Table 1 (*continued*)

Symbol	Description
	Variables
c	control c;
S	the source – the initial state before any attack is launched;
s	source node in an attack graph corresponding to an attack type;
C_k	consequence k;
T	auxiliary sink target node;
$T_{i,k}$	target node of attacker type i causing consequence k;
x	security portfolio x;
x_{cl}	binary indicator of control c at level l;
$y_{ik,e}$	binary indicator whether attacker type i selects edge e to form an attack path from s to the target node $T_{i,k}$.
$D(x)$, $ID(x)$	direct and indirect cost of security portfolio x for the provider;
$I_m(x)$	total indirect cost of portfolio x for user type m;
$L(x)$	security risk for the provider given x;
Payoff^m_{user}	payoff of user type m;
$\text{Payoff}^m_{provider}$	payoff of the provider w.r.t. user type m;
$p_e(x)$	overall probability of a successful attack step associated with edge e given x;
$p_{i,e}(x)$	overall probability of a successful attack step associated with edge e and attacker type i, given security portfolio x;
$\rho_{ik,e}$	dual variable for edge e w.r.t. attacker type i targeting consequence k;
$r_{i,k}(x)$	highest probability that attacker type i will reach consequence C_k given x;
$R_k(x)$	highest probability that consequence k has been breached given x;
$UL_m(x)$	security risks for user type m, given x;

the set of edges; $\overline{h}, \underline{t}$ are functions that return *head* and *tail* of an edge; p is a function returning success rate of exploiting actions (later see Eq. (2)); and $s, \mathcal{T}$ are the source node and the set of target nodes. The source node s in an attack graph is the state where the corresponding attack type initialises attacks. For example, node 0 in Fig. 3(b). The target nodes $\mathcal{T}$ represent specific systems or resources (i.e. privileged states) that an attacker may attempt to compromise or gain access to. Once these targets are compromised, it leads to the corresponding consequences. Please notice these target nodes $\mathcal{T}$ are different from the (unique) sink node T, where all consequence nodes lead to. For example, in the example illustrated in Fig. 3, once attacker type 1 reaches node 3, it can cause consequence 1 and consequence 2. These two consequence nodes connect to the sink node T.

3.2. Security portfolios

The provider seeks the optimal security portfolio to protect the smart home product and users. A security portfolio includes traditional defensive controls to mitigate security risks for both the product and users. For example, access control, account management, patch management, antimalware and anti-phishing software, physical security, etc. They can be found in any traditional cybersecurity framework (e.g. CIS Controls (Center for Internet Security, 2021)). In addition, we consider novel defensive controls to mitigate, in particular, AdvML attacks. For example, robust statistics, robust training and differential privacy, etc. They are based on studies in Pitropakis et al. (2019) and ATLAS[2] proposed by the MITRE Association (The MITRE Corporation. MITRE ATLAS, 2021). In Section 5, we will instantiate these controls as part of our case study and explain how CROSS selects them in a way that optimises the payoff of the provider and the user in the different stages implemented.

The set of controls is denoted by $\mathcal{C}$, and each control can be implemented at different intensity levels signifying the degree of risk reduction. We let $\mathcal{L}(c)$ denote the set of intensity levels of control c and a security portfolio can be expressed using binary indicators x_{cl}, where $c \in \mathcal{C}$ denotes a security control (traditional or AdvML) and $l \in \mathcal{L}(c)$ denotes its intensity level.

If control c at intensity level l is selected, then $x_{cl} = 1$; otherwise $x_{cl} = 0$. Since at most one intensity level of a security control can be implemented, as this level represents a unique implementation of the control, the sum of all levels of a control is less than or equal to one. We thus express a security portfolio as follows:

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

A control can affect multiple edges of the attack graph, meaning that it can stop an attack at different stages of it, and an edge can be augmented by multiple controls, which mitigate security risk at this edge in a combined way. We thus let $\mathcal{C}(e)$ denote the subset of effective controls on edge e. By effective, we refer to controls that can be applied to this edge, positively impacting this edge by reducing its security risk.

We let $\pi_e \leq 1$ represent the baseline success probability of the attacker on unprotected edge e (i.e. when no control is applied on e). Implemented security controls will further reduce the probability of a successful attack step associated with that edge, and the attacker must defeat all implemented controls to succeed finally. This is based on the assumption of independence for security controls. This assumption leads to the multiplicative form in Eq. (2) due to the distinct control mechanisms of different security controls. However, for controls with a degree of correlation, the equation can be extended to incorporate them by introducing a new control. Such an assumption has been justified in Khouzani et al. (2019). Please refer to Khouzani et al. (2019) for details. Thus, the overall probability of a successful attack step associated with edge e, when the provider is defended by the security portfolio x, is denoted by

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

where p_{ecl} is the effectiveness of control c at level l on edge e. Please note that baseline probabilities and effectiveness of controls can be estimated using threat intelligence datasets and surveyed data. For example, effectiveness coefficients are estimated in Schilling and Werners (2016), and Aksu et al. (2017) provides a formulation to estimate the baseline probabilities using Common Vulnerability Scoring System (CVSS).

Since security controls are not cost-free, the provider has to consider: (i) the monetary investment required to implement and maintain a security portfolio; and (ii) the negative impacts on the provider when implementing this portfolio, e.g. patch management may lead to significant downtime for the provider. The direct and

[2] A knowledge base of AdvML tactics, techniques, and real-world case studies published by the MITRE association.

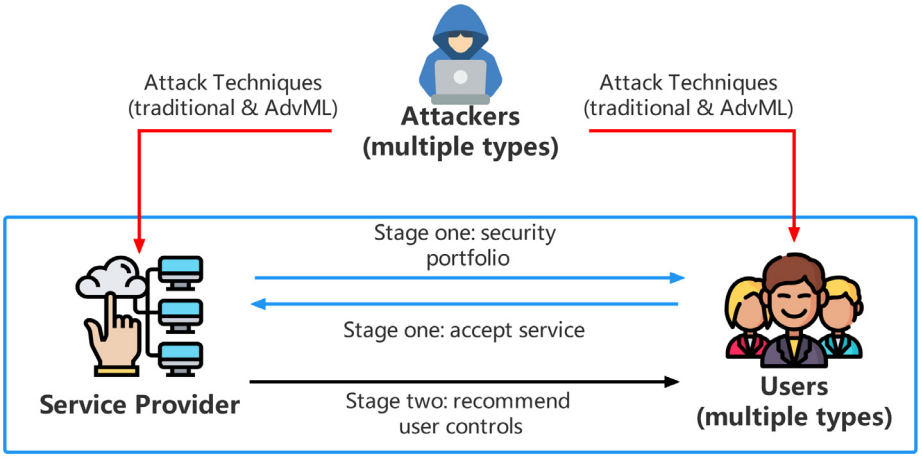

Fig. 2. A multi-leader-follower game.

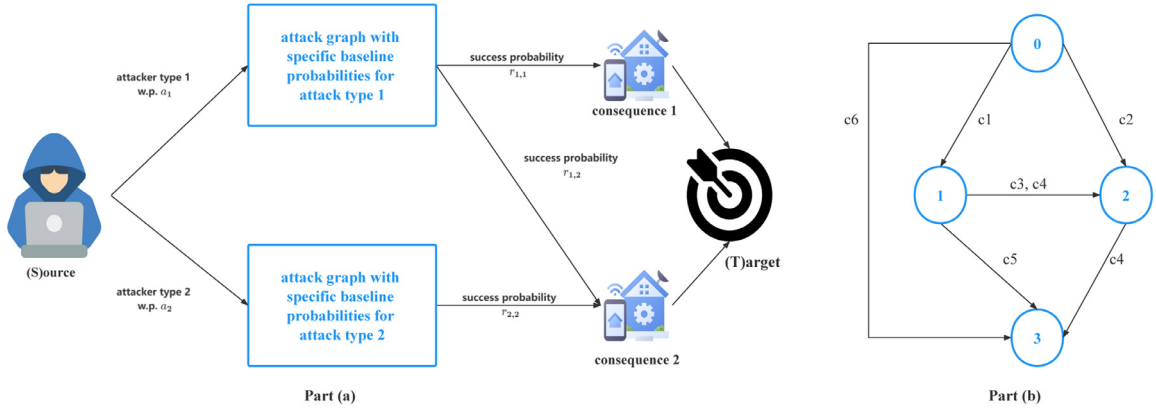

Fig. 3. (a): Example model; (b): Attack graph of the example.

indirect costs of the provider are defined as:

$$D(x) = \sum_{c\in\mathcal{C}, l\in\mathcal{L}(c)} x_{cl}\texttt{Cost}_{\texttt{cl}}, \tag{3}$$

$$ID(x) = \sum_{c\in\mathcal{C}, l\in\mathcal{L}(c)} x_{cl}\texttt{IndirectCost}_{\texttt{cl}}, \tag{4}$$

where $\texttt{Cost}_{\texttt{cl}}$ and $\texttt{IndirectCost}_{\texttt{cl}}$ quantify the direct cost and the indirect cost of control c at level l, respectively.

Recall that a successful attacker may cause multiple consequences to both the provider and the user. We quantify the provider's security risks associated with these consequences denoted by $L(x)$, and derived as follows:

$$L(x) = \sum_{k=1}^{K} R_k(x) \times \texttt{Loss}_k. \tag{5}$$

Recall that $R_k(x)$ represents the highest probability of the attacker causing consequence k out of K consequences given an implemented security portfolio x, and $\texttt{Loss}_k$ quantifies a direct loss associated with consequence k.

3.3. Two-stage defence problem

We consider the defence problem as a two-stage optimisation problem illustrated in Fig. 2. In stage one, the problem is modelled as a multi-leader-follower game.

The provider is the single leader deciding the optimal security portfolio that minimises the security risk $L(x)$ subject to the budget constraints as well as the maximal profit constraints.

The attackers and the users are both followers who do not directly compete with each other. Although the attackers will attack both the provider and the users, the users do not select any security control in stage one but will decide whether to accept the service based on the control selection of the provider. In stage two, the optimisation will select controls for the users to implement.

The (rational) attacker will attack the weakest path from the source to the target to maximise its success probability based on its capabilities to penetrate the defence. In addition, a security portfolio can mitigate propagated security risks for the users; however, as mentioned, it also impacts the user's QoE. Some users may prefer good QoE; whereas others may want a high-security level. Thus the users need to decide whether to accept the service according to the security portfolio. The provider cooperates with the users to find a security portfolio that satisfies that user, so as to maximise the profits from the users.

In stage-two optimisation, CROSS will recommend an optimal portfolio of user controls as an extra layer of security for the users. Apart from the provider controls found in stage-one optimisation, these user controls are security countermeasures that require the user's commitment to applying. In estimating the effectiveness of such a control, we need to take into account the likelihood that the user will simply not follow the recommendation.

3.4. An example

We now provide an example to demonstrate the concepts of the framework and to help the reader to understand the two-stage

Table 2
Effectiveness of controls.

Controls	c_1	c_2	c_3	c_4	c_5	c_6
Effectiveness	*Medium*	*Medium*	*High*	*High*	*Low*	*Low*

Table 3
Effectiveness of user controls.

Controls	uc_1	uc_2	uc_3	uc_4	uc_5	uc_6
Effectiveness	*Medium*	*Low*	*Medium*	*Low*	*Medium*	*Low*

optimisation process[3]. The scenario is shown in Fig. 3 (a), where each rectangle represents the attack graph in Fig. 3 (b). We assume two attacker types with occurrence probability $a_1 = 0.6$ and $a_2 = 0.4$. In the example, attacker type 1 can cause two consequences C_1 and C_2 (node 3 of type 1 user can lead to node C_1 and C_2); while attacker type 2 can only cause consequence C_2 (node 3 of type 2 user only leads to node C_2). Suppose each control has a direct cost of 1 and an indirect cost of 1. For the provider, consequence C_1 has a direct loss $\texttt{Loss}_1 = 20$, and consequence C_2 has a direct loss $\texttt{Loss}_2 = 10$.

Suppose attack path $0 \to 2 \to 3$ includes attack steps that exploit one or more AI functionalities of the provided service, while the other edges are associated with conventional cyber-attacks. Let's assume that attacker type 1 is an expert in performing attacks on the AI service offered by the provider. Thus, attacker type 1 has a higher baseline probability to exploit the edges associated with these attacks successfully: for attacker type 1 we hence set baseline probabilities $\pi_{0\to2}(1) = 0.9$, $\pi_{2\to3}(1) = 0.9$ and $\pi_e(1) = 0.5$ for all other edges e. Attacker type 2 is assumed to be an overall expert of both conventional cyber-attacks and attacks on the AI service: we set the baseline probabilities $\pi_e(2) = 0.8$ for all edges. We use the notations $Low = 0.7$, $Medium = 0.5$, and $High = 0.2$ to represent three fixed levels of control effectiveness. The effectiveness of controls is in Table 2:

While minimising the service security risk, the provider also aims to maximise profits. Thus, the provider wants as many users as possible to use the service.

Recall that a user decides whether to accept the service based on both the security risk and QoE: we let $u_{I,m}$, $u_{UL,m}$ denote the user type m tolerance thresholds for indirect costs (reduction of QoE) and security risks. Furthermore, we set the indirect cost for each control equal to be 1 for each user type and the user loss of consequence 1 and 2 to be 10 and 20, respectively, for each user type. Notice that the loss of the user is different from the direct loss of the provider.

Let's consider two types of users:

- user type one prefers good QoE to high security: we let $u_{I,1} = 2$. There are $U_1 = 10$ type one users, and each pays an average premium of $p_1 = 10$, hence the total possible profit is $P_1 = 100$.
- user type two prefers high security to good QoE: we set $u_{UL,2} = 7.5$. Similarly, there are $U_2 = 10$ type two users, and each pays an average premium of $p_2 = 10$, hence the total possible profit is $P_2 = 100$ too.

Below we illustrate the two-stage optimisation for this particular example. The optimisation below is mathematically justified later on in Section 4.

3.4.1. Stage-one optimisation – Pareto-front solutions

We set the target profit for the provider to be 100, $\overline{P} = 100$, i.e. the provider is willing to provide the service only if the profit is greater than 100. Hence the provider needs to satisfy at least one user type to provide his service.

To compute the Pareto-front, we set the provider's indirect budget B_{ID} to be large and vary the direct cost budget B_D in the range from 1 to 6. The Pareto-front is presented in Fig. 4 (a)-(c).

[3] For the notations used in this section, please refer to Table 1. Model formal details are provided in Section 4.

When $B_D = 1$, the optimisation for the provider selects the portfolio $[c_4]$, and the impact on QoE of both user types is 1. Hence, user type 1 is satisfied. However, due to a high-security risk, user type 2 is not satisfied, resulting in a total profit for the provider of 100. When the direct budget increases to 2 ($B_D = 2$), the optimal portfolio is $[c_4, c_6]$, which further reduces the security risks. In this case, user type 2 becomes satisfied as the security risk is now less than the threshold, resulting in a total profit of 200. Next, when the direct budget is larger ($B_D \geq 3$), the selected optimal portfolio is $[c_1, c_4, c_6]$, which minimises the security risk but introduces a high impact on QoE. User type 1 becomes not satisfied, and the total profit reduces back to 100.

3.4.2. Stage-two optimisation

We now consider stage-two optimisation when stage-one optimisation has selected portfolio $[c_1, c_4, c_6]$. In this case, only user type 2 is satisfied, resulting in a total profit of 100.

We consider six user controls: uc_1 and uc_2 on edge $0 \to 3$, uc_3 and uc_4 on edge $1 \to 3$, and uc_5 and uc_6 one edge $2 \to 3$. Please notice that these controls are not shown on the edges in Fig. 3(b). Each control has a direct and an indirect cost equal to 1, and the effectiveness of user controls is in Table 3:

For user type 2 who prefers high security to good QoE, we set a larger user indirect cost budget, where $\overline{B}_{I,2} = 6$. To compute the Pareto-front solutions, we let the direct costs for each user be in the range of 1 to 6. The impacts on QoE and the loss for user types one and two are presented in Fig. 4 (d). When user direct budget $\overline{B}_{D,2} = 1, 2$, the optimal security portfolios are $[uc_1]$ and $[uc_1, uc_3]$, respectively. Next, with a larger direct cost budget, $\overline{B}_{D,2} = 3$, the user type 2 who prefers high security has a more secure portfolio $[uc_1, uc_2, uc_3]$ to further reduce security risks. When $\overline{B}_{D,2} \geq 4$, the optimal security portfolio is $[uc_1, uc_2, uc_3, uc_5]$.

4. Optimality analysis

In this section, we present the optimality analysis. Apart from minimising the security risk, the provider also wants to maximise the profit from the users. Thus, let's first focus on the Stackelberg game between the provider and the different user types. The result leads to a system of constraints that ensure the maximal profit from the users. Next, we solve stage-one optimisation to find an optimal security portfolio for the provider, given the maximal profit constraints. Finally, we solve stage-two optimisation to find an optimal security portfolio for the users.

4.1. Stackelberg game between the users and the provider

The provider's goal of maximal profit can be modelled as a Stackelberg game between the provider and the users, without considering the attacker. The provider is the leader who first selects a security portfolio that reduces the security risks while affecting the user's QoE. The users are followers who will decide whether to accept the service according to their preferences for security risks and QoE, which are determined by the implemented security portfolio.

Let's distinguish all user types based on their preferences. Let M be the total number of user types. The overall impacts of a security portfolio x on the user's QoE, which represents the user's indirect

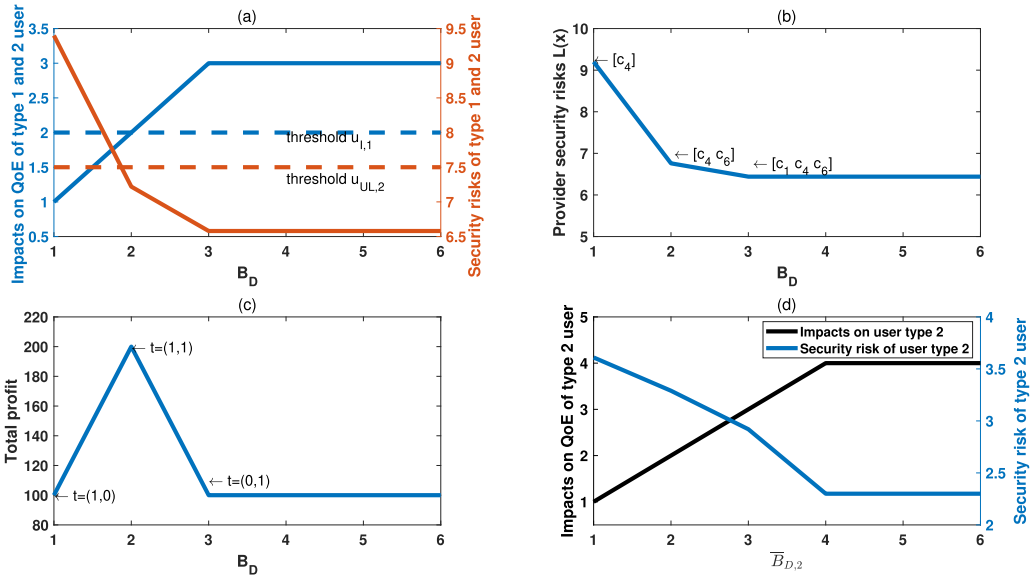

Fig. 4. (a)-(c): Pareto-front for stage-one optimisation; (d): Pareto-front for stage-two optimisation.

cost experienced by the user type m, are defined as follows:

$$I_m(x) = \sum_{c\in\mathcal{C}, l\in\mathcal{L}(c)} x_{cl}\,\texttt{IndCost}_{\texttt{clm}}, \tag{6}$$

where $\texttt{IndCost}_{\texttt{clm}}$ is the user indirect cost (impacts on QoE) imposed to user type m from the selection of control c at implementation level l.

Moreover, if the attacker has successfully exploited the provider, these attacks can propagate to the users. We express the type m user's security risks as follows:

$$UL_m(x) = \sum_{k=1}^{K} R_k(x) \times \texttt{userLoss}_{km}. \tag{7}$$

where $\texttt{userLoss}_{km}$ quantifies the loss for user m when consequence k occurs. Note that $\texttt{userLoss}_{km} = 0$, if the attacks cannot propagate to user type m by exploiting consequence k.

The equations above define a user preference in terms of the impacts on QoE and how much security risk to tolerate. To model this the user's decision-making challenge, recall $u_{I,m}$ and $u_{UL,m}$ are the thresholds which are the highest indirect cost and security risks that user type m is willing to tolerate:

$$I_m(x) \le u_{I,m}, \tag{8}$$

$$UL_m(x) \le u_{UL,m}. \tag{9}$$

These thresholds should be decided based on the user's preference: a user type m who prefers good QoE will have a low threshold $u_{I,m}$, and a user type who prefers a low-security risk will have a low threshold $u_{UL,m}$.

While using the service, user type m has a positive payoff $\text{Payoff}^m_{user}(x) > 0$ if and only if both (8) and (9) hold, i.e. user type m accepts the security risks and can tolerate the impacts on QoE. Otherwise, user type m has a negative payoff. If the number of user type m who sign up with the offered service is denoted by U_m, and the provider gains an average of p_m service premium per user, then the total payoff of the provider is $P_m = U_m \times p_m$ from user type m. If user type m rejects the service (i.e. at least one of (8) and (9) does not hold), the payoffs of that user type and the provider are zero.

Let $\text{Payoff}^m_{provider}(x)$ denote the payoff of the provider gained from user type m. If user type m has a positive payoff (i.e. user type m accepts the offered service), then $\text{Payoff}^m_{provider}(x) = P_m$; otherwise $\text{Payoff}^m_{provider}(x) = 0$, i.e. no gain.

Here the objective of the provider is to find an optimal security portfolio that maximises the sum of payoffs:

$$\max_x \sum_{m=1}^{M} \text{Payoff}^m_{provider}(x), \ \text{s.t.: (1)}. \tag{10}$$

Note that the maximisation here acts as a condition later in stage-one optimisation to guarantee the maximal profit for the provider. The security risks and other objectives will be explicitly addressed in Section 4.2.

The provider aims to maximise his payoff. We let binary indicators t_m for $m = 1, \ldots, M$: $t_m = 1$ indicate whether user type m will accept the offered service; otherwise $t_m = 0$. This payoff maximisation problem can be expressed as follows:

$$\max_{x,t} \sum_{m=1}^{M} t_m \times P_m, \tag{11}$$

$$\text{s.t.: (1)},$$

$$(I_m(x) \le u_{I,m} + (1 - t_m) \times \sigma, \tag{12}$$

$$UL_m(x) \le u_{UL,m} + (1 - t_m) \times \sigma), \ \forall m = 1, \ldots, M. \tag{13}$$

where $\sigma > 0$ is a large number.

For user type m, constrains (12) and (13) are equivalent to (8) and (9) only if $t_m = 1$; otherwise, constrains (12) and (13) are always satisfied. Also if $t_m = 1$, then (8) and (9) must be satisfied. In other words, $t_m = 1$ ensures that user type m will accept the offered service.

Next, we convert the maximisation above into a system of maximal profit constraints. Let $\overline{P} > 0$ be the provider's target, i.e. the provider is willing to provide the service only if the profit can be greater than or equal to the target $\overline{P}$. The objective function can be expressed as follows

$$\exists t_m, m = 1, \ldots, M \ : \ \sum_{m=1}^{M} t_m \times P_m \ge \overline{P}, \tag{14}$$

and (1), ((12), (13), $\forall m = 1, \ldots, M$) hold. Later, in stage-one optimisation, we use these maximal profit constraints to guarantee an optimal security portfolio with at least $\overline{P}$ profit.

Remark 1. Notice that if no user accepts the service, it means that the provider will not make any profit from the users, because no user is satisfied with the QoE or the security risk. In such a scenario, the service provider should re-evaluate the service it provides or the survey data on user preferences.

4.2. Stage-one optimisation

The last section focused on the Stackelberg game between only the provider and the users. Here we add the attacker to address the stage-one defence problem of finding an optimal security portfolio for the provider.

Recall a consequence can be caused by multiple attacker types, and $R_k(x)$ denotes the highest probability of causing consequence k given an implemented security portfolio x. We let $\mathcal{A}_k$ denote the set of all attacker types that can cause consequence k. Thus $R_k(x) = \max_{i\in\mathcal{A}_k}\{r_{i,k}(x)\}$, or equivalently:

$$R_k(x) = \min_{\tau_k} \ \tau_k, \quad \text{s.t.: } \tau_k \geq r_{i,k}(x), \ \forall i \in \mathcal{A}_k. \tag{15}$$

where τ_k is an auxiliary variable.

Stage-one optimisation is as follows:

$$\min_{x,t} L(x), \tag{16}$$

$$\text{s.t.: } (1), ((12), (13), \forall m = 1, \ldots, M), (14),$$
$$D(x) \leq B_D; \ ID(x) \leq B_{ID}, \tag{17}$$

$$R_k(x) \geq r_{i,k}(x), \forall i \in \mathcal{A}_k, k = 1, \ldots, K. \tag{18}$$

Recall (1) represents a security portfolio, and (12), (13) and (14) are maximal profit constraints that ensure there is at least $\overline{P}$ profit for the provider. Moreover, B_D and B_{ID} are the budgets for direct costs and indirect costs. Constraint (17) ensures that the selected security portfolio does not exceed budgets. Finally, (18) is transformed from (15).

4.3. Solving stage-one optimisation

Recall $r_{i,k}(x)$ represents the highest probability that the attacker type i will reach consequence C_k, which can be explicitly expressed as follows:

$$r_{i,k}(x) = \max_{\omega_{s\to T_{i,k}}} a_i \times \prod_{e\in\omega_{s\to T_{i,k}}, c\in\mathcal{C}(e), l\in\mathcal{L}(c)} \pi_e(i)(p_{ecl}x_{cl} + 1 - x_{cl}), \tag{19}$$

where a_i is the occurrence probability of attacker type i, $\pi_e(i)$ is the baseline success probability of attacker type i on edge e, and $T_{i,k} \in \mathcal{T}$ is the target node in the attack graph of attacker type i causing consequence C_k, and $\omega_{s\to T_{i,k}}$ is a path from the source s to $T_{i,k}$.

Note that (19) is a non-linear maximisation problem. Here we give a high-level summary of how to linearise and dualise this problem as presented in Khouzani et al. (2019). We first translate the optimisation variable $\omega_{s\to T_{i,k}}$ into new binary variables $y_{ik,e} \in \{0, 1\}$ for $e \in \mathcal{E}$. For each edge e, a binary variable $y_{ik,e}$ represents whether the attacker type i selects that edge to form an attack path. Equation (19) can be equivalently expressed as follows:

$$r_{i,k}(x) = \max_{y_{ik,e}, e\in\mathcal{E}} a_i \prod_{e\in\mathcal{E}} (p_{i,e}(x)y_{ik,e} + 1 - y_{ik,e}), \tag{20}$$

$$p_{i,e}(x) = \pi_e(i) \prod_{c\in\mathcal{C}(e), l\in\mathcal{L}(c)} (p_{ecl}x_{cl} + 1 - x_{cl}), \tag{21}$$

subject to the linear flow conversion constraints and the binary constraints. Please refer to Lemma 1 in Khouzani et al. (2019). Note that the flow conversion constraints will ensure that the attack path is from the source node s to the target node $T_{i,k}$.

The logarithm function, $\log(x)$, is strictly monotone for $x > 0$. Thus we can convert the product in (20) into a sum:

$$\log(r_{i,k}(x)) = \log(a_i) + \max_{y_{ik,e}, e\in\mathcal{E}} \sum_{e\in\mathcal{E}} \log(p_{i,e}(x)y_{ik,e} + 1 - y_{ik,e}). \tag{22}$$

As $y_{ik,e}$ is a binary variable: $\log(p_{i,e}(x)y_{ik,e} + 1 - y_{ik,e}) = 0$ if $y_{ik,e} = 0$; else if $y_{ik,e} = 1$, then $\log(p_{i,e}(x)y_{ik,e} + 1 - y_{ik,e}) = \log(p_{i,e}(x))$. Thus, the sum can be further reduced to

$$\log(r_{i,k}(x)) = \log(a_i) + \max_{y_{ik,e}, e\in\mathcal{E}} \sum_{e\in\mathcal{E}} y_{ik,e} \log(p_{i,e}(x)). \tag{23}$$

Similarly, $\log(p_{i,e}(x))$ can be translated into a linear function:

$$\log(p_{i,e}(x)) = \log(\pi_e(i)) + \sum_{c\in\mathcal{C}(e), l\in\mathcal{L}(c)} x_{cl} \log(p_{ecl}), \tag{24}$$

Because of Lemma 2 (totally unimodular matrices) in Khouzani et al. (2019), the maximisation problem in (23) can be relaxed to a linear programming (LP) in which the binary constraints (i.e. $y_{ik,e} \in \{0, 1\}$) are equivalent to $y_{ik,e} \geq 0$ for all $e \in \mathcal{E}$.

Next, following Khouzani et al. (2019), we can dualise the maximisation problem into a minimisation problem using strong duality in LP:

$$\log(r_{i,k}(x)) = \log(a_i) + \min_{\rho_{ik}} (\rho_{ik,s} - \rho_{ik,T_{i,k}}), \tag{25}$$

subject to

$$\rho_{ik,\underline{t}(e)} - \rho_{ik,\overline{h}(e)} \geq \sum_{c\in\mathcal{C}(e), l\in\mathcal{L}(c)} \log(p_{ecl})x_{cl} + \log(\pi_e(i)), \forall e \in \mathcal{E}. \tag{26}$$

where ρ_{ik} is a vector of dual variables for the attacker type i targeting consequence k maximisation problem. The detailed conversions can be found in Khouzani et al. (2019).

Thus, constraint (18) is equivalent to

$$\log(R_k(x)) - \log(a_i) \geq \min_{\rho_{ik}} \{\rho_{ik,s} - \rho_{ik,T_{i,k}} : (26) \text{ holds}\} \tag{27}$$

for all $i \in \mathcal{A}_k$, and $k = 1, \ldots, K$. This can be further relaxed to

$$\exists \rho_{ik} : \log(R_k(x)) - \log(a_i) \geq \rho_{ik,s} - \rho_{ik,T_{i,k}}, \tag{28}$$

and (26) holds, $\forall i \in \mathcal{A}_k, \ k = 1, \ldots, K$.

Next, we convert the problem into a standard MICP. Let $z_k = \log(R_k(x)) + \log(\texttt{Loss}_k)$. Then the objective function can be expressed as a sum of exponential functions:

$$L(x) = \sum_{k=1}^{K} R_k(x) \times \texttt{Loss}_k = \sum_{k=1}^{K} \exp(z_k). \tag{29}$$

Definition 1. The exponential cone is a convex subset of $\mathbb{R}^3$ MOSEK ApS (2020):

$$\mathcal{K}_{exp} = \mathbf{cl}\{(x_1, x_2, x_3) : x_1 \geq x_2 \exp(x_3/x_2), \ x_2 > 0\}. \tag{30}$$

Let λ_k, for $k = 1, \ldots, K$, be optimisation variables. Then, we can exactly convert the problem into a MICP with exponential cones:

$$\min_{x, \{\rho_{ik}, i\in\mathcal{A}_k, k=1,\ldots,K\}, \lambda, z, t} \sum_{k=1}^{K} \lambda_k, \tag{31}$$

$$\text{s.t.: } (1), ((12), (13)^*, \forall m = 1, \ldots, M), (14), (17),$$
$$(z_k - \log(\texttt{Loss}_k) - \log(a_i) \geq \rho_{ik,s} - \rho_{ik,T_{i,k}}, \tag{32}$$

$$(26), \forall i \in \mathcal{A}_k), \ \forall k = 1, \ldots, K,$$
$$(\lambda_k, 1, z_k) \in \mathcal{K}_{exp}, \forall k = 1, \ldots, K. \tag{33}$$

Proposition 1. *In (13)*, $R_k(x)$ are replaced by $\lambda_k/\texttt{Loss}_k$ for $k = 1, \ldots, K$.*

Proof. The conic constraint (33) is equivalent to the inequality $\lambda_k \geq \exp(z_k) = R_k \times \texttt{Loss}_k$. Thus, we have $\lambda_k - v_k = R_k \times \texttt{Loss}_k$ where $v_k \geq 0$ is a slack variable. In (13), for user type m and $t_m = 1$, the inequality constraint $\sum_{k=1}^{K} R_k(x) \times \texttt{userLoss}_{km} \leq u_{UL,m}$ is equivalent to

$$\sum_{k=1}^{K} \lambda_k/\texttt{Loss}_k \times \texttt{userLoss}_{km} \leq u_{UL,m} + \sum_{k=1}^{K} v_k/\texttt{Loss}_k \times \texttt{userLoss}_{km}.$$

Thus, if a tighter inequality constraint $\sum_{k=1}^{K} \lambda_k/\texttt{Loss}_k \times \texttt{userLoss}_{km} \leq u_{UL,m}$ holds, then $\sum_{k=1}^{K} R_k \times \texttt{userLoss}_k \leq u_{UL,m}$ also holds. Note that (13) always holds if $t_m = 0$.

Furthermore, the proof of Proposition 2 later shows that an optimal solution must have $v_k^* = 0$ for all $k = 1, \ldots, K$. Thus, $\sum_{k=1}^{K} R_k \times \texttt{userLoss}_k \leq u_{UL,m}$ and $\sum_{k=1}^{K} \lambda_k/\texttt{Loss}_k \times \texttt{userLoss}_{km} \leq u_{UL,m}$ are equivalent in the defence problem. □

Proposition 2. *The conic constraint (33) is equivalent to the inequality $\lambda_k \geq \exp(z_k)$. Minimising $\sum_{k=1}^{K} \lambda_k$ subject to the conic constraints is equivalent to minimise $\sum_{k=1}^{K} \exp(z_k)$.*

Proof. The inequality $\lambda_k \geq \exp(z_k)$ is equivalent to $\lambda_k - v_k = \exp(z_k)$ where $v_k \geq 0$ is the slack variable, which is the same as in Proposition 1. As a result, we have $\min \sum_{k=1}^{K} \lambda_k = \min(\sum_{k=1}^{K} \exp(z_i) + v_k)$. Since a minimisation solution must have $v_k^* = 0$, we have $\lambda_k^* = \exp(z_k^*)$. Finally, since for all k, $\lambda_k \geq \exp(z_k)$, we have $\min \sum_{k=1}^{K} \lambda_k = \min \sum_{k=1}^{K} \exp(z_k)$. □

The provider must be able to address multiple attacker types. In stage-one optimisation, a_i denotes the probability that the attacker is of type i. The defence problem considers both the success probability $r_{i,k}$ with the occurrence probability a_i associated with an attacker of type i when finding an optimal security portfolio. This is motivated by considering the case of a powerful attacker type who is highly likely to exploit the provider successfully. However, the provider may be less likely to face such a powerful attacker type, i.e. a low a_i.

If the provider is interested in the worst-case scenario, we can let $a_i = 1/N$ for all $i = 1, \ldots, N$. In that case, the optimisation will focus on those attacker types with the highest success probability.

4.4. Stage-two optimisation

While stage-one optimisation focuses on the provider, stage-two optimisation selects optimal portfolios of user controls to provide each user type with an extra layer of security. These user controls need to be implemented by the user, not the provider.

One may argue about combining the two stages into one single optimisation. In such an optimisation, the optimal security portfolio would then include both types of controls, i.e. the provider and user controls. However, the provider cannot force the users to implement these user controls. If a non-cooperative user refuses to implement the recommended user controls, the whole security portfolio determined in the single optimisation scenario becomes not optimal, resulting in increased security risks. For this reason, we split the problem into two stages: stage-one optimisation finds the default optimal portfolio for the provider itself and all user types. Next, for each user type, stage-two optimisation seeks an optimal security portfolio of user controls to help further improve security for that user type.

Since the following variables in stage-two optimisation are similar to those in stage-one optimisation, we do not repeat them in Table 1.

4.5. Solving stage-two optimisation

First, let binary variables $\bar{x}_{cl}$ express a portfolio of user controls: $\bar{x}_{cl} = 1$ indicates user control c at level l is selected; otherwise $\bar{x}_{cl} = 0$. The set of user controls is denoted by $\bar{\mathcal{C}}$.

Note that stage-two optimisation focuses on further reducing propagated security risks to the users. Hence, we only consider those consequences that can propagate security risks to the users, and let $\bar{\mathcal{K}}$ denote the set of those consequences. For example, we have three possible consequences (C_1, C_2 and C_3) considered in stage-one optimisation, and only consequence C_2 may propagate risks to the users. Note that all user types face the same set of possible consequences. Then $\bar{\mathcal{K}} = \{C_2\}$. In other words, $\texttt{userLoss}_{km} > 0$ for $k \in \bar{\mathcal{K}}$.

Similarly, we let $r_{i,k}(\bar{x}; x)$ be the highest success probability associated to attacker type i given the provider security portfolio x, expressed as follows:

$$r_{i,k}(\bar{x}; x) = \max_{\omega_{s \to T_{i,k}}} a_i \prod_{e \in \omega_{s \to T_{i,k}}, c \in \bar{\mathcal{C}}, l \in \mathcal{L}(c)} \bar{\pi}_e(i; x) \times (\bar{p}_{ecl}\bar{x}_{cl} + 1 - \bar{x}_{cl}) \tag{34}$$

where $\bar{\pi}_e(i; x)$ represents the baseline success probability of attacker type i on edge e given the security portfolio x, and $\bar{p}_{ecl}$ is the effectiveness of control c at level l on edge e. Thus, the highest probability to cause consequence $C_k \in \bar{\mathcal{K}}$ is denoted as $R_k(\bar{x}; x) = \max_{i \in \mathcal{A}_k}\{r_{i,k}(\bar{x}; x)\}$.

Because of varying preferences for security and QoE, user types have different budgets for user controls. Next, we let $\bar{B}_{D,m}$ and $\bar{B}_{I,m}$ denote the direct cost budget and the indirect cost budget for user type m.

Notice that the indirect cost budget $\bar{B}_{I,m}$ is the threshold of the highest impacts on QoE user type m will tolerate when adding user controls. It depends on the applied security portfolio found in stage-one optimisation

Thus, the stage-two optimisation problem for user type m is as follows:

$$\min_{\bar{x}} UL_m(x) = \sum_{k \in \bar{\mathcal{K}}} R_k(\bar{x}; x) \times \texttt{userLoss}_{km}, \tag{35}$$

$$\text{s.t.: } \bar{x}_{cl} \in \{0, 1\}, \forall c \in \bar{\mathcal{C}}, l \in \mathcal{L}(c); \quad \sum_{l \in \mathcal{L}(c)} \bar{x}_{cl} \leq 1, \forall c \in \bar{\mathcal{C}}, \tag{36}$$

$$\sum_{c \in \bar{\mathcal{C}}, l \in \mathcal{L}(c)} \bar{x}_{cl}\overline{\texttt{Cost}}_{clm} \leq \bar{B}_{D,m}; \quad \sum_{c \in \bar{\mathcal{C}}, l \in \mathcal{L}(c)} \bar{x}_{cl}\overline{\texttt{InCost}}_{clm} \leq \bar{B}_{I,m}; \tag{37}$$

$$R_k(\bar{x}; x) \geq r_{i,k}(\bar{x}; x), \forall i \in \mathcal{A}_k, k \in \bar{\mathcal{K}}. \tag{38}$$

where represents a security portfolio of user controls, ensures the direct and indirect budgets for user type m are not exceeded, and (38) is converted from $R_k(\bar{x}; x) = \max_{i \in \mathcal{A}_k}\{r_{i,k}(\bar{x}; x)\}$.

Next, we convert the problem into a MICP. Here we only provide a sketch of the conversions because stage-two optimisation is similar to stage-one optimisation. Using strong duality and totally unimodular matrices, we transform the maximisation problem (38) into a minimisation problem. Then we convert the nonlinear optimisation into a tractable MICP with exponential cone constraints, which is expressed as follows:

$$\min_{\bar{x}, \{\bar{\rho}_{ik}, i \in \mathcal{A}_k, k \in \bar{\mathcal{K}}\}, \bar{\lambda}, \bar{z}} \sum_{k \in \bar{\mathcal{K}}} \bar{\lambda}_k, \tag{39}$$

s.t.: (36), (37);

$$(\bar{\lambda}_k, 1, \bar{z}_k) \in \mathcal{K}_{exp}, \forall k \in \bar{\mathcal{K}}, \tag{40}$$

$$(\bar{z}_k - \log(\texttt{userLoss}_{km}) - \log(a_i) \geq \bar{\rho}_{ik,s} - \bar{\rho}_{ik,T_{i,k}}, \tag{41}$$

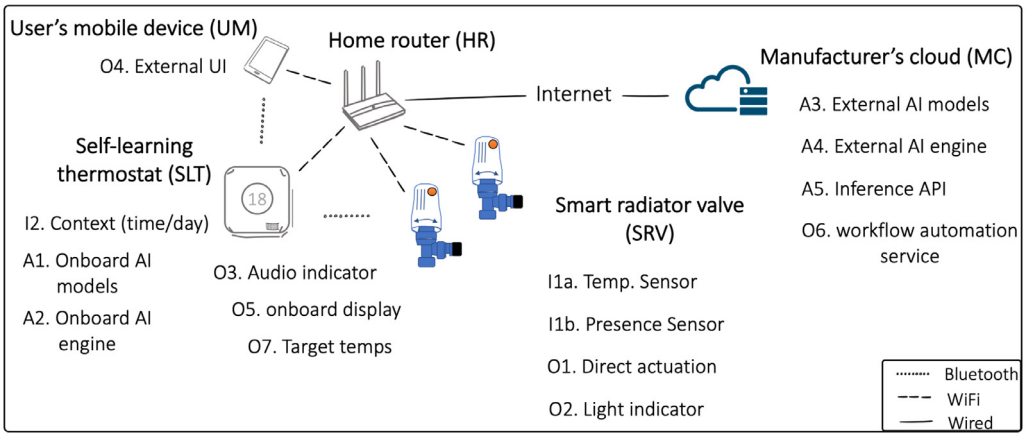

Fig. 5. Configuration of AI-enabled smart heating.

$$\overline{\rho}_{ik,\underline{t}(e)} - \overline{\rho}_{ik,\overline{h}(e)} \geq \sum_{c\in\overline{c}(e),l\in\mathcal{L}(c)} \log(\overline{p}_{ecl})\overline{x}_{cl} + \log(\overline{\pi}_e(i;x)), \forall e \in \mathcal{E}), \forall i \in \mathcal{A}_k, k \in \overline{\mathcal{K}}, \quad (42)$$

where $\overline{z}_k = \log(\overline{R}_k(\overline{x};x)) + \log(\texttt{userLoss}_{km})$ and $\overline{\lambda}_k$ are auxiliary variables, and $\overline{\rho}_{ik}$ are dual variables.

5. Case study

This case study models a prototypical AI-enabled Internet of Things (IoT) environment that combines characteristics and vulnerabilities currently present on existing commercial off-the-shelf devices. The purpose of this case study is twofold: (i) to assess the performance and recommendations of CROSS to the provider of an AI service and (ii) to shed light on the trade-offs emerging when protecting AI services deployed in smart homes from not only traditional cyber attacks but also AdvML ones, which can be available to the attacker when the traditional security controls have been bypassed. In this use case, we focus on the user side; hence we ignore several threats and controls which are more specific to the enterprise security of the provider.

The provider is a manufacturer/seller of an AI-enabled heating technology for smart homes. Their product is a self-learning thermostat (SLT) coupled with smart radiator valves (SRV). These devices were manufactured to offer an individual, room-by-room heating control, by operating in conjunction with a thermostat kit, and they can be easily managed through a smartphone application. The provider's objective is to design a security control strategy that mitigates the associated cybersecurity risks while maximising financial profit. Fig. 5 illustrates the scenario modelled.

The product is able to learn the user's temperature preferences in different rooms and automatically adjust the target temperature in each one (O7) based on whether they are present or not at home (I1b) and other context (I2), such as time of day and energy price. The training of the machine learning models of SLT, as well as firmware updates, require connection to the manufacturer's cloud (A3, A4), but a subset of smart features, such as the human presence detection in each room happens on the SRV.

Based on this scenario, we have developed a simple attack graph (Fig. 6), taking into account known vulnerabilities in similar commercial off-the-shelf devices. Table 4 presents the attack steps used in the attack graph. Please note that *SP* represents the service provider and *O1* represent the direct actuation of the smart radiator valve, which turns the valve on or off. Node 0 represents the source when the system operates normally (`nor`). Node 1 is the state where the home router (HR) has been exploited. Edge $0 \rightarrow 1$ represents several possible attack actions, including **BRUTE_HR, AUTH_HR**, and **SE_HR**. Nodes 6 and 7 represent integrity breaches (`int`): node 6 is an integrity breach on the cloud and 7 on the SRV. Edge $1 \rightarrow 6$ combines several attack actions forming a ML poisoning attack path: **IDOR_SLT, PUB_MAT, SE_MC, INF_API**, and **POISON_ML**. Edge $1 \rightarrow 7$ represents two possible attack actions: **DREB_SLT**, and **DREB_SRV**. Nodes 5 and 4 are the privilege states where the provider's and the SRV's availability have been breached (`ava`). Edge $0 \rightarrow 4$ combines several BT (Bluetooth) attack actions forming a BT attack path: **BT_SLT, BT_SN** or **BT_SM**. Moreover, edge $0 \rightarrow 5$ represents a DoS (Denial of Service) attack to MC i.e. **DOS_MC**. Once the provider has been breached, the security risk can propagate to the device (i.e. edge $6 \rightarrow 7$ and $5 \rightarrow 4$). Node 2 represents a privilege state where the SLT has been exploited through a BT attack (edge $0 \rightarrow 2$). Next, two attack paths, `insider` attack and `Evasion-ML` attack, lead to node 3 where the provider's confidentiality has been breached (`con`). `Evasion-ML` attack is formed by multiple attack actions, including **PUB_MAT, SE_MC, INF_API, REP_MOD**, and **ATT_INF**. Possible security controls, including both traditional and AdvML for these attacks are added in Fig. 6; details of their costs and effectiveness are in Table 5. Notice that `Ed-U` and `2FA-U` are user controls, which should only be selected in stage two optimisation. Since we assume the direct budget of the provider is large later in stage-one optimisation (see Section 5.1), we omit the direct cost of controls. The indirect costs of controls for the provider and the user are quantified in a range of `VL(VeryLow)`, `L(Low)`, `M(Medium)`, `H(High)`, `VH(VeryHigh)`, representing costs 1, 2, 3, 4, 5, respectively. Similarly, the effectiveness has five levels: $VL(VeryLow) = 0.9$, $L(Low) = 0.7$, $M(Medium) = 0.5$, $H(High) = 0.3$, and $VH(VeryHigh) = 0.1$.

5.1. Stage-one optimisation

Here we consider a single attacker type (i.e. $a_1 = 1$) who can cause all five possible consequences (see Fig. 6). Moreover, we consider one user type 1 who prefers good QoE to a low-security risk (i.e. a large $u_{UL,1}$). Notice that the provider must satisfy user type 1.

The provider's direct losses with respect to consequence C_1 to C_5 are 4, 2, 8, 15, 6, respectively. These values are educated guesses, which are meant to reflect the idea that if the service provider experiences a compromise, it would result in a large number of users being affected and therefore incurring large losses for the service provider, and relatively small losses for the individual user. These

Table 4
Attack actions.

Attack	Description	Attack	Description
ATT_INF	Already knowing the model, the attacker uses the inference API to try to derive the most likely current value of human presence with an attribute inference attack	**AUTH_HR**	Due to improper restriction of excessive authentication attempts, an attacker in range can recover the PIN and access the network (e.g., CVE-2021-20635).
BRUTE_HR	Brute force attack for gaining access to home WiFi network.	**BT_SLT**	Discover Bluetooth SLT devices in the vicinity that should normally be undiscoverable (e.g., CVE-2020-15802). Known more widely as BLURtooth attack.
BT_SM	The Bluesmacking attack uses the L2CAP layer to transfer an oversized packet to a Bluetooth device for the purpose of denial of service. (e.g., CVE-2006-3146)	**BT_SN**	The Bluesnarfing attack involves exploiting the OBEX protocol to transfer information from Bluetooth devices.
DREB_SLT	DNS Rebinding attack exploiting SLT: Expose API via internal network with no authentication. Then, interact with the API to alter target temperatures on SLT (e.g., CVE-2018-11315).	**DREB_SRV**	DNS Rebinding attack exploiting SRV: Expose API via internal network with no authentication. Then, interact with the API to alter target temperatures on SRV (e.g., CVE-2018-11315).
DOS_MC	A conventional denial of service attack on the servers of the Provider.	**IDOR_SLT**	Exploit system's insecure direct object references vulnerability allowing user-supplied input to access objects directly (e.g., CVE-2020-8791).
INF_API	Identify and access inference API provided to craft and test different adversarial examples.	**POISON_ML**	Data poisoning attack at the server so as to affect all users of an AI device.
PUB_MAT	Search for publicly available material, such as whitepapers and publications on algorithms used by the developers with the aim to understand the AI engine employed.	**REP_MOD**	By repeatedly querying the inference API, an attacker can replicate the machine learning model as shown by Wallace et al. (2020).
SE_MC	Social engineering attacks targeting access on the servers of the Provider.	**SE_HR**	Social engineering attack targeting access on the User's home router.

Table 5
Costs and effectiveness of security controls.

Control	Parameters	Descriptions	Control	Parameters	Descriptions
Security Controls – Parameters = ($\texttt{IndirectCost}_{\texttt{cl}}$, $\texttt{IndCost}_{\texttt{cl,1}}$, Effectiveness)					
(Ed-S, 1)	(VL, 0, L)	Staff training on protection against social engineering attacks delivered once per year. Its effectiveness is found to drop noticeably after the first six months (Reinheimer et al., 2020).	(Ed-S, 2)	(M, 0, M)	Staff training on protection against social engineering attacks delivered every four months, which is the point up to which staff are still found to be as effective at spotting attacks as just after the previous training iteration (Reinheimer et al., 2020).
(Ed-U, 1) (user control)	(0, VL, VL)	Low-commitment security awareness activity recommended to the users, for example in the form of a video (Heartfield et al., 2016)	(Ed-U, 2) (user control)	(0, L, L)	High-commitment security training recommended to the users, for example in the form of a game (Hart et al., 2020).
(2FA-S, 1)	(L, 0, H)	Standard two-Factor Authentication (2FA) for the provider's staff, for example through mobile text message or app.	(2FA-S, 2)	(H, 0, VH)	Advanced 2FA for the provider's staff, e.g. physically unclonable function based two-factor authentication (Gope and Sikdar, 2018).
(2FA-U, 1) (user control)	(0, M, H)	Standard two-Factor Authentication (2FA) for the user access to the SLT and SRV, for example through mobile text message or app.	(2FA-U, 2) (user control)	(0, VH, VH)	Advanced 2FA for the user access to the SLT and SRV, e.g. physically unclonable function based two-factor authentication (Gope and Sikdar, 2018).
(Pa-S, 1)	(L, VL, L)	Patching policy of long provisioning time (e.g. 30 days) to minimise strain on company resources, but with increased risk of attacks prior to patch deployment and associated reputational damage (Morgner et al., 2020).	(Pa-S, 2)	(M, L, M)	Patching policy of short provisioning time (e.g. 10 days) to minimise risk of attacks prior to patch deployment and associated reputational damage, but at the cost of increased strain on company resources (Morgner et al., 2020).
(FnT, 1)	(L, 0, M)	Stateful firewall for network traffic filtering.	(FnT, 2)	(M, 0, H)	Advanced firewall, e.g. allowing detection of intrafirewall policy anomaly rules (Togay et al., 2021).
(DoS, 1)	(L, 0, M)	IoT DoS mitigation for the provider's servers, using low-cost deployment, such as IoT Honeypot-based (Anirudh et al., 2017).	(DoS, 2)	(M, 0, H)	Advanced IoT DoS mitigation for the provider's servers, e.g. IoT-Middelware (Sicari et al., 2018) or SDN (Ahmed and Kim, 2017).
(DLP-strat, 1)	(L, 0, M)	Data loss prevention strategy prioritising insider threat management (Alneyadi et al., 2016).			
Adversarial Machine Learning Controls– Parameters = ($\texttt{IndirectCost}_{\texttt{cl}}$, $\texttt{IndCost}_{\texttt{cl,1}}$, Effectiveness)					
(RoS, 1)	(L, L, M)	AI Robust Statistics that use constraints and regularisation techniques to reduce potential distortions of the learning model caused by poisoned data (Tabassi et al., 2019).	(RoI, 1)	(L, L, M)	Robust Improvement including techniques such as "Adversarial Training", "Gradient Masking", "Defensive Distillation", "Ensemble Method", "Feature Squeezing", and "Reformers/Autoencoders" (Tabassi et al., 2019).
(DiP, 1)	(H, M, M)	Differential privacy (Tabassi et al., 2019).	(PuB, 1)	(M, 0, L)	Detailed restrictions on publications, limiting information to primarily high-level descriptions of datasets and AI models and techniques.
(PuB, 2)	(H, 0, M)	Organisation-wide ban on publication of datasets and AI models and techniques.			

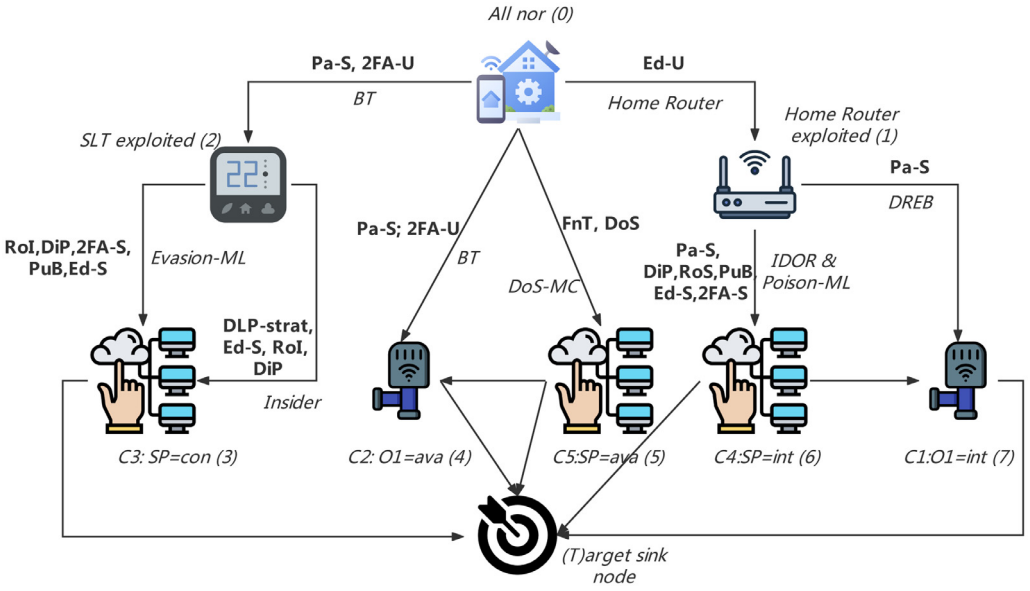

Fig. 6. case study – attack graph.

values are used to illustrate the optimisation solutions. In practice, these values could be obtained through a survey conducted by the service provider. As previously noted, there are several examples of surveys, such as the *NCSC*'s survey on the UK's Cyber Security Breaches in 2022 (The National Cyber Security Centre, 2022), *Which*'s report on smart home cybersecurity, and *Kaspersky*'s report on IoT attacks. Nonetheless, the service providers should perform their own surveys based on their specific services and users.

To study the Pareto-front solutions, we will assume the direct budget of the provider is large, and we will vary the indirect budget and the threshold of the highest indirect cost $(B_{ID}, u_{I,1})$ for the provider and user type 1 in the range of 0 to 30. We also assume the baseline probability of each edge to be 0.9. Below, we present some significant solutions on the Pareto-front:

- When the indirect budget the threshold are small, i.e. $(B_{ID}, u_{I,1}) = (3, 3)$, the optimisation will first protect those consequences with a high direct loss, i.e. C_4: SP=int and C_3: SP=con. The optimal portfolio consists of controls `(Ed-S, 1)` and `(2FA-S, 1)`, resulting in the security risk of 17.53 with $R_1 - R_5$=[0.81, 0.9, 0.57, 0.17, 0.90]. So far, consequences C_4: SP=int and C_3: SP=con have been protected by effective controls.
- When $(B_{ID}, u_{I,1}) = (8, 8)$, the optimal portfolio adds controls `(Pa-S, 1)` and `(DoS, 1)`, resulting in the security risk of 8.76 with $R_1 - R_5$=[0.41, 0.45, 0.28, 0.09, 0.45]. Control `(Pa-S, 1)` and `(DoS, 1)` can mitigate potential attacks on BT, DREB and DoS attacks. So far, every consequence has been protected by at least one control.
- As the indirect budget threshold increases up to $(B_{ID}, u_{I,1}) = (16, 16)$, the optimal portfolio upgrades the existing control to a high level and also adds AdvML controls. For example, at $(B_{ID}, u_{I,1}) = (16, 16)$, the security portfolio is `(Ed-S, 1)`, `(2FA-S, 2)`, `(FnT, 2)`,`(Pa-S, 2)`, `(RoI, 1)`, `(DoS, 2)`, which reduces the security risk to 4.56 with $R_1 - R_5$=[0.41, 0.45, 0.14, 0.028, 0.081].
- As the indirect budget and the threshold further increase, more controls are added to the security portfolio. At $(B_{ID}, u_{I,1}) = (30, 30)$, the security portfolio includes all the provider controls with the highest intensity level, which results in the security risk of 3.25 with $R_1 - R_5$=[0.41, 0.45, 0.03, 0.004, 0.08].

5.2. Stage-two optimisation

In stage two, the provider can select a portfolio of user controls to provide the user with an extra layer of security. Suppose, in stage-one optimisation, the indirect budget and the threshold are $(B_{ID}, u_{I,1}) = (26, 15)$. The optimal security portfolio is `(Ed-S, 2)`, `(2FA-S, 2)`, `(FnT, 2)`,`(Pa-S, 2)`, `(RoI, 1)`, `(DoS, 2)`, `(DiP, 1)`, `(RoS, 1)`, `(DLP-strat, 1)`, resulting in a security risk of 3.28 with $R_1 - R_5$=[0.41, 0.45, 0.025, 0.0051, 0.081]. Moreover, $I_1 = 9 < u_{I,1}$. Thus, we have $\overline{B}_{I,1} = 6$ for stage-two optimisation.

Suppose, for example, both user controls `Ed-U` and `2FA-U` have direct-cost 0, i.e. the user does not need to pay for these two user controls, and the user's loss with respect to consequences C_1 and C_2 are both 5. Consequences C_3 to C_5 only impact the service provider and are therefore not taken into account in the second stage optimisation. In stage-two optimisation, the provider will recommend user controls `(Ed-U, 2)` and `(2FA-U, 1)` resulting in the security risk of 2.09 with $R_1 - R_2$=[0.2835, 0.135].

6. Conclusion

We proposed a novel decision support system to help service providers and users select optimal portfolios of security controls to counteract cyber attacks in the smart home supply chain. The proposed system considers the important roles of both the service provider and the users in determining optimal security portfolios and utilises a multi-objective bi-level two-stage optimization approach, where the first stage focuses on the role of the service provider in securing devices, and the second stage focuses on the user. The system also incorporates financial and security constraints, and different user and attacker profiles. We demonstrated the effectiveness of the proposed system through a case study of an AI-powered smart radiator valve.

We noticed that the optimisation results depend on the parameters of the user and the attacker profiles. Therefore, inaccurate data may result in sub-optimal security portfolios. In addition, our assumption of complete rationality of both users and attackers may not always hold in real-world scenarios, as users may deviate from the best response and undermine the optimal defence strategy. Moreover, partially rational attackers may affect the sur-

vey data and reduce the effectiveness of the optimal security portfolio against rational attackers. Furthermore, our game assumes cooperative users. However, in reality, some users may not cooperate and may even deactivate controls proposed by the provider. Future work should address this behaviour and consider the inclusion of a regulator or governmental body to ensure security standards are met. Thus, future work will focus on validating the framework in real-world scenarios experiments, collecting data, and testing the system in different smart home environments and devices, with different rationality of user and attacker profiles, to evaluate its performance, adaptability, and robustness. The objective is to improve the feasibility and usefulness of the proposed decision support system in real-world smart home supply chains.

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.

CRediT authorship contribution statement

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

Data availability

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

Acknowledgment

This research was supported by the EPSRC grants (EP/T026596/1 and EP/T026812/1) under the “CHAI: Cyber Hygiene in AI enabled domestic life” project.

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. Association for Computing Machinery, New York, NY, USA, p. 378392.

Ahmed, M.E., Kim, H., 2017. DDoS attack mitigation in internet of things using software defined networking. In: 2017 IEEE Third International Conference on Big Data Computing Service and Applications. IEEE, pp. 271–276.

Aksu, M.U., Dilek, M.H., Tatlı, E.I., Bicakci, K., Dirik, H.I., Demirezen, M.U., Aykır, T., 2017. A quantitative CVSS-based cyber security risk assessment methodology for IT systems. In: 2017 International Carnahan Conference on Security Technology (ICCST). IEEE, pp. 1–8.

Alneyadi, S., Sithirasenan, E., Muthukkumarasamy, V., 2016. A survey on data leakage prevention systems. Journal of Network and Computer Applications 62, 137–152.

Anirudh, M., Thileeban, S.A., Nallathambi, D.J., 2017. Use of honeypots for mitigating dos attacks targeted on IoT networks. In: 2017 International Conference on Computer, Communication and Signal Processing. IEEE, pp. 1–4.

Biggio, B., Roli, F., 2018. Wild patterns: ten years after the rise of adversarial machine learning. Pattern Recognit 84, 317–331.

Bispham, M.K., Agrafiotis, I., Goldsmith, M., 2019. Nonsense attacks on google assistant and missense attacks on amazon alexa. In: International Conference on Information Systems Security and Privacy.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., Zhou, W., 2016. Hidden voice commands. In: 25th USENIX Security Symposium. USENIX Association, Austin, TX, pp. 513–530.

Cavusoglu, H., Raghunathan, S., Yue, W.T., 2008. Decision-theoretic and game-theoretic approaches to IT security investment. Journal of Management Information Systems 25 (2), 281–304.

Center for Internet Security. 2021. CIS critical security controls v8. https://www.cisecurity.org/controls/; (accessed November 10, 2021).

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., Mukhopadhyay, D., 2021. A survey on adversarial attacks and defences. CAAI Transactions on Intelligence Technology 6 (1), 25–45.

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

Do, C.T., Tran, N.H., Hong, C., Kamhoua, C.A., Kwiat, K.A., Blasch, E., Ren, S., Pissinou, N., Iyengar, S.S., 2017. Game theory for cyber security and privacy. ACM Comput Surv 50 (2), 1–37.

Durkota, K., Lisy, V., Kiekintveld, C., Bosansky, B., 2015. Game-theoretic algorithms for optimal network security hardening using attack graphs. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, p. 17731774.

Fang, F., Nguyen, T.H., Pickles, R., Lam, W.Y., Clements, G.R., An, B., Singh, A., Tambe, M., Lemieux, A., 2016. Deploying PAWS: field optimization of the protection assistant for wildlife security. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press, p. 39663973.

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

Gope, P., Sikdar, B., 2018. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. IEEE Internet Things J. 6 (1), 580–589.

Gordon, L.A., Loeb, M.P., 2002. The economics of information security investment. ACM Transactions on Information and System Security (TISSEC) 5 (4), 438–457.

Gu, T., Dolan-Gavitt, B., Garg, S., 2017. Badnets: identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:170806733.

Hart, S., Margheri, A., Paci, F., Sassone, V., 2020. Riskio: a serious game for cyber security awareness and education. Computers & Security 95, 101827.

Heartfield, R., Loukas, G., Gan, D., 2016. You are probably not the weakest link: towards practical prediction of susceptibility to semantic social engineering attacks. IEEE Access 4, 6910–6928.

Jain, M., Kiekintveld, C., Tambe, M., 2011. Quality-bounded solutions for finite bayesian stackelberg games: Scaling up. In: The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, p. 9971004.

Khouzani, M., Liu, Z., Malacaria, P., 2019. Scalable min-max multi-objective cyber-security optimisation over probabilistic attack graphs. Eur 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: European Symposium on Research in Computer Security. Springer International Publishing, Cham, pp. 179–197.

Li, Y., Xu, L., 2020. Cybersecurity investments in a two-echelon supply chain with third-party risk propagation. Int. J. Prod. Res. 1–23.

Liu, Q., Li, P., Zhao, W., Cai, W., Yu, S., Leung, V.C.M., 2018. A survey on security threats and defensive techniques of machine learning: a data driven view. IEEE Access 6, 12103–12117.

Lubin, M., 2017. Mixed-integer convex optimization: outer approximation algorithms and modeling power. Massachusetts Institute of Technology.

Morán R, D.A., Dey, S.S., Vielma, J.P., 2012. A strong dual for conic mixed-integer programs. SIAM J. Optim. 22 (3), 1136–1150.

Morgner, P., Mai, C., Koschate-Fischer, N., Freiling, F., Benenson, Z., 2020. Security update labels: establishing economic incentives for security patching of iot consumer products. In: 2020 IEEE Symposium on Security and Privacy. IEEE, pp. 429–446.

Panda, S., Panaousis, E., Loukas, G., Laoudias, C., 2020. Optimizing investments in cyber hygiene for protecting healthcare users. In: From Lambda Calculus to Cybersecurity Through Program Analysis. Springer, pp. 268–291.

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. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, p. 895902.

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.

Pitropakis, N., Panaousis, E., Giannetsos, T., Anastasiadis, E., Loukas, G., 2019. A taxonomy and survey of attacks against machine learning. Computer Science Review 34, 100199.

MOSEK ApS. Modeling cookbook 3.2.2. https://docs.mosek.com/modeling-cookbook/expo.html; 2020.

The National Cyber Security Centre. Cyber Security Breaches Survey 2022. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2022/cyber-security-breaches-survey-2022; 2022.

PYMNTS. Kaspersky detects 1.5b IoT cyberattacks this year. https://www.pymnts.com/news/security-and-risk/2021/kaspersky-detects-iot-cyberattacks-double-last-year/; 2021.

Reinheimer, B., Aldag, L., Mayer, P., Mossano, M., Duezguen, R., Lofthouse, B., von Landesberger, T., Volkamer, M., 2020. An investigation of phishing awareness and education over time: When and how to best remind users. In: Sixteenth Symposium on Usable Privacy and Security, pp. 259–284.
Sanjab, A., Saad, W., Başar, T., 2017. Prospect theory for enhanced cyber-physical security of drone delivery systems: A network interdiction game. In: 2017 IEEE international Conference on Communications (ICC). IEEE, pp. 1–6.
Sawik, T., 2013. Selection of optimal countermeasure portfolio in IT security planning. Decis Support Syst 55 (1), 156–164.
Sawik, T., 2020. A linear model for optimal cybersecurity investment in industry 4.0 supply chains. Int. J. Prod. Res. 1–18.
Sawik, T., 2021. Balancing cybersecurity in a supply chain under direct and indirect cyber risks. Int. J. Prod. Res. 1–17.
Schilling, A., Werners, B., 2016. Optimal selection of IT security safeguards from an existing knowledge base. Eur J Oper Res 248 (1), 318–327.
Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., Meyer, G., 2012. Protect: A deployed game theoretic system to protect the ports of the united states. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. Citeseer, pp. 13–20.
Sicari, S., Rizzardi, A., Miorandi, D., Coen-Porisini, A., 2018. Reato: reacting to denial of service attacks in the internet of things. Comput. Networks 137, 37–48.
Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. arXiv preprint arXiv: 13126199.
Tabassi, E., Burns, K., Hadjimichael, M., Molina-Markham, A., Sexton, J., 2019. A taxonomy and terminology of adversarial machine learning. NIST IR.
Togay, C., Kasif, A., Catal, C., Tekinerdogan, B., 2021. A firewall policy anomaly detection framework for reliable network security. IEEE Trans. Reliab..
The MITRE Corporation. 2021. MITRE ATLAS. https://atlas.mitre.org/.
The MITRE Corporation. 2022. MITRE att&ck matrix for enterprise. https://attack.mitre.org/matrices/enterprise/.
Vaněk, O., Yin, Z., Jain, M., Bošanský, B., Tambe, M., Pěchouček, M., 2012. Game-theoretic resource allocation for malicious packet detection in computer networks. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, p. 905912.
Wallace, E., Stern, M., Song, D., 2020. Imitation attacks and defenses for black-box machine translation systems. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 5531–5546.
Which. How a smart home could be at risk from hackers. https://www.which.co.uk/news/article/how-the-smart-home-could-be-at-risk-from-hackers-akeR18s9eBHU; 2021.
Yan, C., Ji, X., Wang, K., Jiang, Q., Jin, Z., Xu, W., 2022. A survey on voice assistant security: attacks and countermeasures. ACM Comput Surv 55 (4), 1–36.
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. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, p. 855862.
Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., Xu, W., 2017. Dolphinattack: Inaudible voice commands. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 103–117.
Zhang, Y., Malacaria, P., 2021. Bayesian stackelberg games for cyber-security decision support. Decis Support Syst 148, 113599.
Zhang, Y., Malacaria, P., 2022. Optimization-time analysis for cybersecurity. IEEE Trans Dependable Secure Comput 19 (4), 2365–2383.
Zonouz, S.A., Khurana, H., Sanders, W.H., Yardley, T.M., 2013. RRE: a game-theoretic intrusion response and recovery engine. IEEE Trans. Parallel Distrib. Syst. 25 (2), 395–406.

Yunxiao Zhang received the B.Eng. degree in Electrical and Electronic Engineering from Newcastle University, UK, and the MSc degree in Control Systems and the Ph.D. degree from Imperial College London, UK. Currently, he is a Postdoctoral Research Assistant at the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK, with Professor Pasquale Malacaria. His current research interests lie in cyber-security investment, optimization, game theory.

Pasquale Malacaria received his Laurea in Philosophy from a 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.

George Loukas is a Professor of Cyber Security and Head of the IoT and Security Centre at the University of Greenwich. He has been project coordinator or principal investigator in seven national and international research projects related to the security of modern digital infrastructures, from robotic vehicles and smart buildings to new virtual reality environments. Prof. Loukas has a Ph.D. from Imperial College (2006). He is on the Editorial Board of IEEE Transactions in Information Forensics and Security and Elsevier Simulation Modelling Practice and Theory.

Emmanouil Panaousis is Professor of Cyber Security at the Univ. of Greenwich, and Founder of the Cyber Risk Lab. He previously held appointments at the Univ. of Surrey, Univ. of Brighton, Imperial College London and Queen Mary Univ. of London. His research expertise is in the area of cybersecurity and privacy risk management. His research is funded by the European Commission, the UK Engineering and Physical Sciences Research Council, and the National Cyber Security Centre.