The Cost of Repetition: A Compositional Scalability Model for Attack Trees*

Clemens Fruböse and Eva Hetzel

KASTEL Security Research Labs, Karlsruhe Institute of Technology, Karlsruhe, Germany
{clemens.fruboese,eva.hetzel}@kit.edu

Abstract. Repeated cyber attacks rarely entail constant attacker costs: Tool reuse, learning effects, detection, and access burn-out can produce economies or diseconomies of scale. Yet existing quantitative attack tree analyses typically treat costs and impacts as static values and therefore miss how attacker incentives change under repetition. In this paper, we add a previously overlooked dimension to attack tree based risk analysis, which is the number of attack executions. We lift cost and damage from static scalars to execution-indexed functions and provide compositional AND/OR propagation rules that yield path-level cost and damage profiles, enabling joint cost–damage analysis under repeated executions. The resulting cost–damage relations can be non-concave due to scalable costs and nonlinear damage effects (e. g., economies of scale, saturation, or threshold behavior).
On illustrative attack trees, we show that scalability reshapes optimal attacker choices across objectives (net benefit, return on investment, budget- and target-constrained): A path optimal for a single execution may be suboptimal for multiple executions, even switching paths across executions can be optimal. This accounts for why attacker strategies evolve with scale. Our framework provides a sound basis to compare such objectives, to anticipate path choices as attack executions increase, and to support quantitative, repeatable security assessments.

Keywords: Attack trees · cumulative cost curves · risk quantification · scalability of attacks · security economics.

1 Introduction

The scalability of an attack is the damage-to-cost ratio between the attacker's costs and the damage resulting from the attack subject to multiple attack executions [13]. Considering the scalability of attacks in risk analysis can reveal

* This is the extended version of the paper published under the same name in the proceedings of the Third International Joint Conference on Quantitative Evaluation of Systems and Formal Modeling and Analysis of Timed Systems (QEST+FORMATS 2026) by Springer Nature. The extended version includes more detailed descriptions in the appendix.

attacks that have high initial costs, but are profitable in the attacker's eyes if executed multiple times. This can occur because of diminishing marginal cost for an additional attack or increased damage when a certain number of assets is affected. For example, in critical infrastructure (such as energy systems), many components are designed to include redundancies to mitigate the impact of a single attack. An attacker would therefore want to attack multiple times to damage the system. Since this important aspect of attacks has been overlooked in previous risk analysis frameworks, we propose extending the cost–damage analysis by the dimension of the number of attacks as it can explain attacker behavior previously deemed unlikely.

Incorporating this additional dimension into the cost–damage comparison renders the cost–damage analysis three dimensional, as depicted in Fig. 1. This view captures the behavior of the cost–damage properties of attack paths under repeated attack executions. We observe that this view reduces to existing cost–damage analysis in the case of one single attack. However, a projection of these foliations onto the cost–damage plane yields a trajectory showcasing the changing cost–damage behavior over attack repetitions. In particular, intersections of these trajectories indicate a shift in the choice of the optimal attack path.

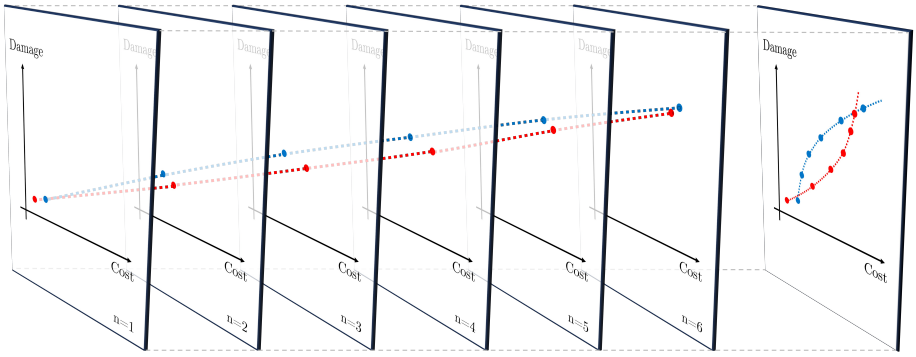


Fig. 1: Three-dimensional foliations of the cost–damage plane, subject to several attack executions. Each surface corresponds to a fixed execution count; big dots mark the discrete cumulative cost–damage points. Connecting the dots across execution counts yields trajectories on the cost–damage plane, revealing how path economics evolve and where optimal choices may switch captured by the notion of scalability of different attacks.

Constructing such trajectories is non-trivial: (i) basic events along a composite path can scale differently; (ii) multi-stage attacks may deliver partial damage before completion; and (iii) assessing costs separately for each n is cumbersome.

To address this, we use attack trees and their natural decomposition to propagate upwards the cost and damage functions of each node. We revisit observations from economics to assess that cost behavior is naturally determined as cost

functions with respect to a given number of executions. Inverting these functions enables to obtain the number of possible attacks given a budget. Using the dependency of damage on successful attacks, we can construct the cost–damage trajectories for each path and investigate the scalabilities as depicted in Fig. 2.

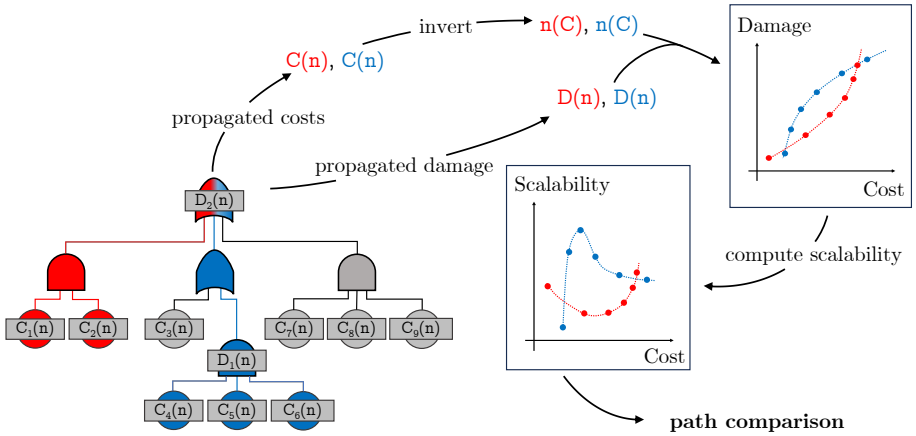


Fig. 2: Procedure overview. Cost and damage functions are propagated along the attack tree to obtain path-level profiles. Their combination yields cost–damage trajectories; the scalability computation highlights efficiency points enabling path comparison.

Our approach formalizes an attacker analyzing the attack space for the most cost-efficient attack: While depending on the goals, the attacker may want to maximize return on investment (ROI) or net profit or optimize with respect to given boundary conditions such as minimal required damage or maximal budget. For example, a company may want to harm a competing company in a cost-efficient way but only if the cost of the attack does not ruin the malicious company, too.

As the notion of scalability relates the damage to the required cost, the choice according to these circumstances can be predicted. As a consequence—based on the expected attacker types—preventive measures to harden the most critical attack paths can be implemented.

1.1 Contributions

We summarize the paper's main contributions as follows:

– **Scalability-aware attack trees.** We lift attack tree quantification from static leaf costs to *execution-indexed cumulative cost functions* $C_{v_i}(n)$ and provide *compositional propagation rules* for AND/OR refinements to obtain path-level cost profiles $C_P(n)$ and their budget-indexed counterparts.

- **Nonlinear attack economics under repetition.** By combining scalable costs with (potentially saturating or threshold) damage, our framework captures *non-concave cost–damage behavior*, implying that attacker optima need not be unique or interior (boundary optima and multiple local optima can arise), in contrast to standard diminishing-returns assumptions.
- **Objective-dependent path reversals.** We connect common attacker objectives (net benefit, ROI, budget/target constraints) to *average vs. marginal scalability* diagnostics and show that scalability induces *path-preference reversals* as budget grows; we illustrate these effects on a multi-stage example and a real-world case study.

1.2 Outline

The paper is structured as follows: After reviewing related work in Sect. 2, setup and propagation rules for cost and damage functions are introduced in Sect. 3.1. Section 3.2 formally defines attack scalability and Sect. 3.3 revisits economies and diseconomies of scale together with cost function classes as well as exemplary damage functions. Section 4 works through an illustrative and a real-world example. Section 5.1 characterizes efficiency points and their existence, leading to a path reversal. Section 5.2 analyzes path switching under repeated executions—demonstrated in Sect. 5.3. Sections 6 and 7 close with a discussion and conclusions.

2 Related Work

Hetzel, Nemes, and Müller-Quade [13] consider the scalability of attacks with a more theoretical approach, where costs are represented by oracle accesses. This oracle can break any assumption the security of a protocol is based on once per access. The impact of an attack is measured in affected units, e. g., participants in an authentication infrastructure or votes in a voting scheme. In this work, we take a more practical approach using attack trees [25]. Our framework can be used to further analyze existing attack trees to improve risk analysis and highlight so far overlooked threats.

Quantifying security using attack trees is a well-established approach [19,23,17]. Additionally, relating the attacker's costs to the resulting damage has been proposed [20,6,2,4]. A common limitation, however, is the absence of a temporal component, which would reveal how optimal choices evolve under repeated executions. Although Jürgenson and Willemson consider time by allowing the attacker to sequence elementary steps to switch to a different path in case of failure, each attack is only conducted once [15]. André, Lime, Ramparison, and Stoelinga add the duration of an attack as an additional parameter to the attacker's costs and the damage of an attack [1]. Here, the temporal component is reflected by *multiple executions*, i. e., repeating entire paths.

Classical work has long recognized nonlinear effects under repetition: Learning/experience and coordination curves yield economies and diseconomies of

scale [29,12,5]. Damage need not be linear either—saturation or redundancies can diminish marginal damage [16,21], whereas in interdependent infrastructures (e. g., power grids) numerous perturbations exceeding a threshold can escalate via cascades [24,7]. These effects have however not been considered in attack tree based risk analysis.

In their seminal article from 2002, Gordon and Loeb propose a defender centric economic model for determining which assets warrant protection investment [11]. In this sense, our paper studies the complementary attacker-side problem: choosing the most cost-effective attack strategy.

3 Preliminaries and Model of Scalability

In this section, we provide the basis for our framework. After a short repetition of attack trees, we present various definitions of attack scalability. Then we provide a collection of cost and damage functions to be plugged into the scalability definitions.

3.1 Propagation of Cost and Damage Along Attack Trees

Attack trees [25] are a widely used tool in risk analysis. Ideally, they are a representation of all possible attack vectors an attacker could take to harm the system that is analyzed. Attack trees contain the attacker's goal in the root node. The internal nodes of the tree can most commonly be either OR or AND nodes, indicating that either one of the child nodes is sufficient to reach the goal or that all of them have to be fulfilled. The leaf nodes represent the so-called elementary attacks, which are not split up any further and are assumed to be independent of each other. While damage of an attack can be assigned to any node, only leaf nodes carry the attacker's costs as proposed by Lopuhaä-Zwakenberg and Stoelinga [20]. Both impact as well as costs are traditionally fixed values [25], which we lift to functions in this paper. The terms impact and damage are used synonymously here.

The different attack paths can be found by identifying the minimal cut sets of the tree. Each minimal cut set consists of elementary attacks which can lead to the attacker's goal if they are combined. Since the cut sets are minimal, the goal cannot be reached anymore if one of the elementary attacks is removed from the set. Looking at the attack paths resulting from the minimal cut sets, all OR connections have been removed from the tree, since at every OR connection, the attacker can take different routes to reach the root node.

To compute the attacker's costs for each path, at AND connections, the costs of the child nodes can simply be added:

$$C_{\text{total}} = \sum_{i=1}^{k} C_{v_i}, \tag{1}$$

where i is the index of one of the k child vertices. In more complex trees, there might be further connections, e. g., where an inner node can be executed by

choosing two out of three elementary attacks [18]. Such connections are treated analogously to OR connections in the minimal cut set analysis. Gates that require the parallel (PAND) or sequential (SAND) execution of attacks are of less interest in our framework, since monetary costs in particular are added at AND gates to compute the costs of the path, independent of the order of execution.

In an attack tree, all paths lead to at least the impact at the root node. Further damage can be caused by internal nodes of the tree, which can be interpreted as collateral damage resulting from intermediate steps leading to the actual goal of the attacker at the root node. Nevertheless, it can also show a nonlinear behavior after multiple attack executions, as discussed in Sect. 3.4.

3.2 Definition of Attack Scalability

Intuitively, the scalability of an attack relates damage and required cost to the number of executions. We quantify scalability through *cost–damage efficiency*, i. e., the ratio of cumulative damage to cumulative cost. We provide both an execution-indexed and a cost-indexed view, since cumulative cost and damage are often given in execution-indexed form, while the cost-indexed view is more natural for economic comparisons.

Setup. Let $C : \mathbb{N}_{\geq 1} \to (0, \infty)$ be the cumulative cost function on execution count n with $C(n+1) > C(n)$ for all $n \geq 1$. Let $D : \mathbb{N}_{\geq 1} \to [0, D_{\max}]$ be the cumulative (monetized) damage function that is non-decreasing and bounded by $D_{\max}$. They assign cost and damage to the of number of executions, respectively.

Definition 1 (Execution-indexed Scalability Ratio[1]). *The execution-indexed scalability ratio of the attack is defined as the function*

$$S(n) \coloneqq \frac{D(n)}{C(n)}, \quad n \geq 1. \tag{2}$$

This definition can be understood as a measure of average cost-efficiency after $n \geq 1$ executions and is the cost-aware relaxation of [13].

As different attacks usually have different costs, attackers are more interested in the impact with respect to costs than with attack executions. Accordingly, we obtain $D\big(n(C)\big)$, damage as function of cost, by inverting the function $C(n)$ to calculate the number of possible attacks given a specific budget:

$$n(C') \coloneqq C^{-1}(C'), \quad C' \geq C_{\min} \coloneqq C(n=1). \tag{3}$$

Definition 2 (Cost-indexed Scalability Ratio). *The cost-indexed scalability ratio of the attack is defined as the function*

$$S(C') \coloneqq \frac{D(n(C'))}{C'}, \quad C' \geq C_{\min} \coloneqq C(n=1). \tag{4}$$

[1] This ratio is mathematically equivalent to the classical benefit–cost ratio (BCR) known from economics, but since we analyze the growth behavior of attacks, we use the term scalability to emphasize the technical interpretation.

These definitions of scalability measure average cost efficiency and hence mask the behavior around a state of interest. Accordingly, the notion of marginal utility gives insight into the current behavior of efficiency and can hence answer whether a further investment is beneficial.

Definition 3 (Marginal Utility). *The marginal utility is defined as*

$$MU(C') \coloneqq \frac{\Delta D\big(n(C')\big)}{\Delta C'}, \;\; C' \geq C_{\min} \coloneqq C(n=1). \tag{5}$$

3.3 Attack Costs Under Repeated Attacks

All cumulative cost functions, denoted by $C(n)$, are strictly monotonically increasing, since upon every attack execution, the attacker has to either invest more or can benefit from previous investments with barely any additional costs. On the other hand, attacks can also become more expensive, e. g., when resources are used up or complexity increases. Accordingly, cost functions may be convex, concave, or neither.

Appendix A presents four cost functions derived from economic laws and their corresponding cumulative cost curves: Wright's law, which describes an increase in experience and efficiency with every produced item [29], Brooks's law, which describes diseconomies of scale due to growing communication overhead in large groups [5], non-rival goods which cannot be exhausted [26], and lastly constant costs if no scaling can be observed. The different types of cost functions are summarized in Table 1 in the top rows. The table also includes some examples for attacks that show the respective cost behavior across attack executions.

3.4 Damage Under Repeated Attacks

Similar to the attacker's costs, the damage of an attack (per attack execution) can change with every time the attack is performed. On one hand, many systems rely on redundant components to ensure availability. While the first attack only leads to a small damage (the cost of replacing or repairing the redundant component without any consequences to the rest of the system) a second attack can cause much more harm.

On the other hand, a system might have higher value and lower value targets. Once an attacker has compromised all higher value targets, the additional damage in further attacks decreases. Another reason for the damage per attack execution to decline might be that full destruction of the system is approached.

Just like the cumulative cost function, the cumulative damage function $D(n)$ is monotonically increasing, since further attacks do not reduce damage. Damage functions can also be convex or concave functions and their domain and range are limited, since once the target is fully destroyed, no more damage is possible. In App. A, two types of damage functions are discussed exemplarily [27,14,28]. They can also be found in Table 1 in the bottom rows.

If independence and linearity of the cumulative damage are assumed for illustration, one may write $D(n) = n \cdot d_1$ with constant per-execution damage

d_1; in general, interactions, saturation, or synergies imply nonlinearity and are captured by the definitions of Sect. 3.2.

Table 1: Examples of cost and damage functions, at the top and bottom of the table respectively

Type	Reasoning	Characterization, Typical Values	Examples
Sublinear [29]	Gained experience and efficiency	$C(n) = c_1 \cdot \big(1 + \frac{1}{1+\log_2(b)} \cdot (n^{1+\log_2(b)} - 1)\big)$, $b \in [0.75, 0.9]$	Malware development for device class, lock picking
Superlinear [5]	Growing communication overhead and training effort	$C(n) = n \cdot \frac{n^2-n}{2}$	Bypassing layered detection systems, coordinated multi-actor attacks
Zeroth Order [26]	Goods used by multiple parties	$C(n) = \varepsilon \cdot n + b$, ε very small	Self-replicating malware, supply chain attacks
First Order	No cost saving	$C(n) = c_1 \cdot n$, c_1 equal to single attack cost	Physical-layer attack (e. g., GNSS jamming, break open a car)
Saturating [27,14]	Few high value targets	$D(n) = D_{\max} \cdot \frac{n^b}{n^b+a}$, $D_{\max}$ total system value, $a = 1 - \frac{D_{\max}}{D(1)}$, b discriminatory power	Peer-to-peer networks with supernodes
Cascading [28]	Interdependent infrastructures	$D(n) = \frac{D_{\max}}{1+\exp(-b(n-n_0))}$, $D_{\max}$ total system value, b growth rate, n_0 executions to damage half	Redundant systems (e. g., electrical grid)

4 Application of the Framework to Examples

In this section, we present two example attack trees: one to showcase our scalability framework and one real-world example. The latter stems from a commissioned security analysis of the recently introduced German electronic patient record.

4.1 Application to a Generic Tree

To illustrate the framework, we first give an example tree in Fig. 3. Looking at the structure of the tree, we find all minimal sets which lead to a damage and obtain three different paths:

$$P1 = \{v_1, v_2, v_7, v_{10}, v_{11}\}, \quad P2 = \{v_3, v_4, v_8, v_{10}, v_{11}\}, \quad P3 = \{v_5, v_6, v_9, v_{11}\}.$$

When we assign cumulative costs and damages to the nodes as indicated in Fig. 3, we can compute the cumulative costs and damages for each path using the propagation rules from Sect. 3:

$$C_{P1}(n) = \frac{1}{40}\frac{n^3 - n^2}{2} + 0.5 + \varepsilon n \qquad D_{P1} = \frac{12}{1 + \exp(-n + 17)} + 35 \cdot \frac{n}{n + 28}$$

$$C_{P2}(n) = 1.5 + \varepsilon n + n, \qquad D_{P2} = \frac{12}{1 + \exp(-n + 17)} + 35 \cdot \frac{n}{n + 28}$$

$$C_{P3}(n) = \frac{5(n^{1+\log_2(0.6)} - 1)}{1 + \log_2(0.6)} + 5 + 2 + \varepsilon n \qquad D_{P3} = 35 \cdot \frac{n}{n + 28} \quad ; \varepsilon > 0.$$

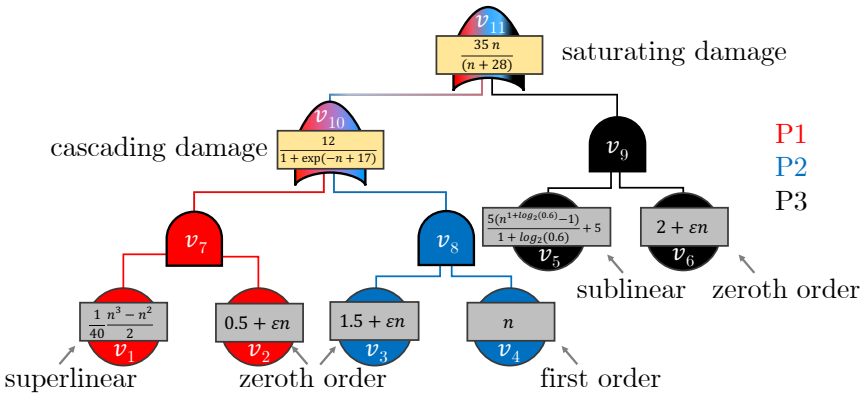
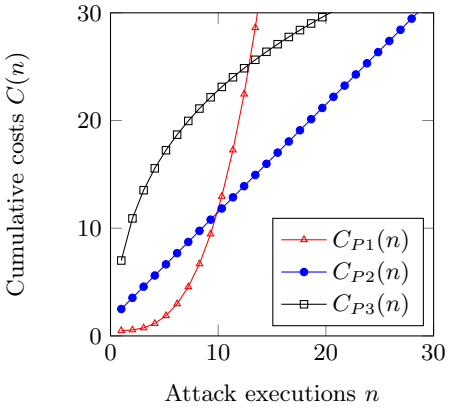


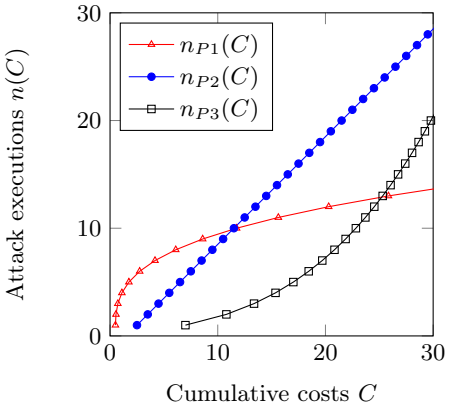
Fig. 3: Example attack tree to illustrate the framework. According to standard notation, AND and OR gates are represented with a round or tapered top, respectively. Each node may carry a damage function (yellow box), while only leaf nodes can carry a cost function (gray box).

The cumulative costs are plotted in Fig. 4a. One can see that the combination of different basic events amounts to the shape of the curves: While for five iterations $P2$ is more expensive than $P3$, this is reversed at their crossing. We then invert the cumulative cost functions using the variable transformation of

Eq. 3 to obtain the number of possible attack executions with respect to budget in Fig. 4b. We only consider cumulative cost greater than the respective $C_{\min} := C(n = 1)$.

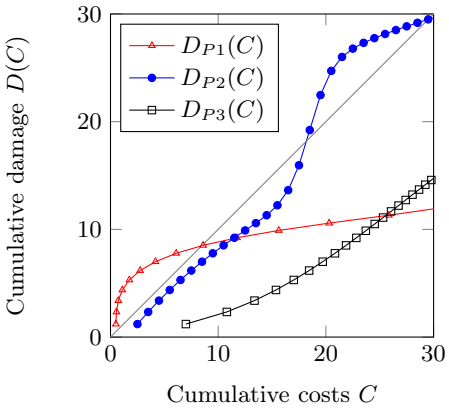


(a) Cumulative cost given attack executions.

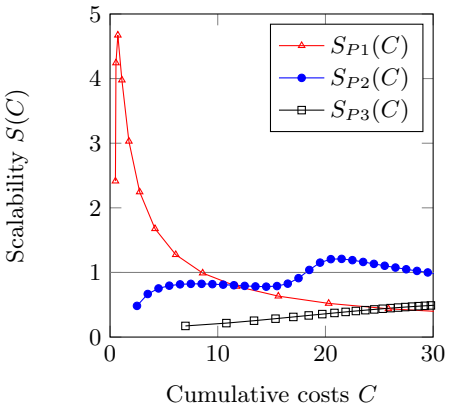


(b) Attainable number of attack executions given cumulative costs.

Fig. 4: Cumulative cost for different attack paths (left) and corresponding possible attack executions given a budget (right). We observe the superlinear, linear and sublinear type. Note their respective axial symmetry along the bisectrix.



(a) Cost–damage plot.



(b) Cost-indexed scalability plot.

Fig. 5: Cost–damage plot (left) and scalability plot (right) for all paths. The thin gray line represents equality of cost and damage, attacks above it are profitable.

By plugging the cumulative cost functions into the cumulative damage functions, we obtain the cost–damage graph shown in Fig. 5a. While useful for some decision problems, Sect. 5.1 argues that the scalability graph better captures different attacker goals. Using Def. 2, we therefore derive the scalability plot in Fig. 5b. A detailed scalability analysis of this example tree follows in Sect. 5.3.

4.2 Electronic Patient Record (ePA for all)

Before the rollout of the electronic patient record ePA in Germany in January 2025, a security analysis using attack trees has been performed [10]. Five attacker goals were identified, leading to five attack tree root nodes: unauthorized reading, manipulating or deleting of a record, denial of service of the record, and revealing treatment/ diseases/ personal data of the insured person. The report assigns success probabilities to attacks and attacker types.

Looking, e. g., at the attack tree with the attacker goal unauthorized reading of a record (partially shown in App. C, Fig. 12) and the group of external attackers like hackers or hacktivists, there are seven different major paths the attacker could take, which again split up into multiple subpaths. The two groups of paths with the highest success rates for this type of attacker are via the nodes "gain access via the primary system of the healthcare provider" ($p = 24\%$) and "gain access to the front end of the insured person" ($p = 59\%$). Therefore, the latter seems to have a higher risk. However, when the attacker gains access to the front end of one user, they can only read this user's patient record. When gaining access to the primary system of the healthcare provider, all of their patients are affected. The success probability of this attack path is smaller, but it seems to scale much better. The report [10] does note that healthcare providers have access to a large number of patient records, but this is not reflected in the classification merely based on success probabilities.

Figure 6 compares the cost-indexed scalability ratio curves of two attack paths, where the red solid line corresponds to an attack via the healthcare provider and the blue dashed line corresponds to an attack via the insured person. Please note the logarithmic scale. With the analysis in our framework we can confirm the intuition that indeed the attack via the healthcare provider scales much better and might be preferred by the attacker in the long run. Once the attacker has invested two units of cost, the solid scalability curve crosses the dashed one and stays at a far higher scalability value. A more extensive analysis of this part of the attack tree on unauthorized reading of a patient record can be found in App. C.

5 Choice of Attack Vector Based on Scalability

After identifying the scalability curves for all attack tree paths, the conclusions of the security analysis are still ambiguous. Different attacker budgets and goals might lead to different path preferences.

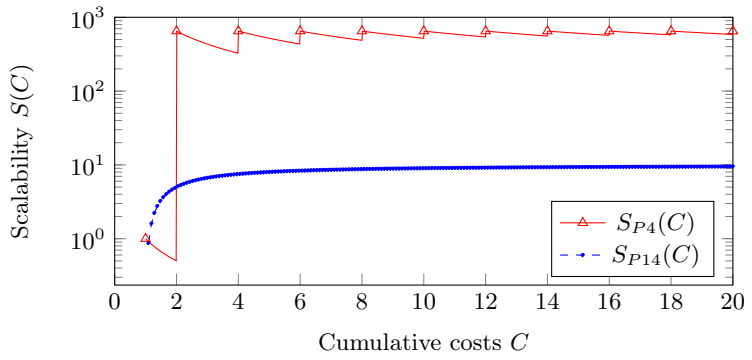


Fig. 6: Selection of two cost-indexed scalability ratio curves of the attack tree on unauthorized reading of a record with $c_1 = 1$ and $D(n) = n$.

Continuous relaxation. We now treat cumulative cost and damage as sufficiently smooth to enable closed-form solutions and existence arguments for efficiency points using basic calculus, in the spirit of classical security-economics models. Accordingly, we will then require that $C : [1, \infty) \to (0, \infty)$ be continuously differentiable, $n \in \mathbb{R}$ with $\mathrm{d}/\mathrm{d}n \; C(n) > 0$ for all $n > 0$ and $D : [1, \infty) \to [0, D_{\max}]$ continuously differentiable, non-decreasing and bounded by $D_{\max}$. We require $n > 1$, $C \geq C_{\min} = C(1)$ to exclude poles of high scalability at $C = 0$.

5.1 Objectives and their Relation to Scalability

Existing work has considered different attacker behavior subject to different economic rationales [11,6,20]. We formalize four natural attacker goals with budget indexing for a path P:

(a) Maximize net profit (NP) as difference between damage and cost. The attacker searches for the best combination of path and cost:

$$\max_{P, C \geq C_{\min,P}} \big(NP_P(C)\big) \coloneqq \max_{P, C \geq C_{\min,P}} \big(D_P\big(n(C)\big) - C\big). \tag{6}$$

(b) Maximize return on investment (ROI). The attacker seeks the best return on investment—defined as the ratio of net benefit by cumulative cost:

$$\max_{P, C \geq C_{\min,P}} \mathrm{ROI}_P(C) \coloneqq \max_{P, C \geq C_{\min,P}} \left(\frac{D_P\big(n(C)\big) - C}{C}\right). \tag{7}$$

(c) Maximize damage subject to a budget. Given a budget $B > 0$, the objective is

$$\max_{P, C_{\min,P} \leq C \leq B} D_P\big(n(C)\big). \tag{8}$$

(d) Minimize cost subject to a required damage. Given a required minimum damage $D_{\min} > 0$, the objective is

$$\min_{P, C \geq C_{\min,P}} \big(C\big) \quad \text{s.t. } D_P\big(n(C)\big) \geq D_{\min}. \tag{9}$$

We note that $D_P(C)$ is continuous, non-decreasing, and bounded above by some $D_{\max}$ as we are using the continuous relaxation to enable closed-form solutions. Hence, both $NP_P(C)$ and $\mathrm{ROI}_P(C)$ reach a maximum in their half-open domain $[C_{\min,P}, \infty)$. However, we did not impose a concavity assumption on D_P. In particular, the interplay of $D(n)$ and $n(C)$ can yield a *non-concave* (e. g., locally convex) function $D\big(n(C)\big)$. Thus, the maxima need not be unique.

For a fixed path P, the four objectives yield the following solution characterizations which again justify the consideration of scalability S and marginal utility MU:

(a) *Profit peak.* Interior stationary points of $NP_P(C)$ satisfy

$$\partial_C \; NP_P(C)\big|_{C=C^\star} = 0 \iff MU_P(C^\star) = 1,$$

i. e., marginal benefit equals marginal cost. Since $D_P(C)$ is not assumed to be concave, such stationary points are only necessary conditions: $C^\star$ may correspond to a local maximum, a local minimum, or a stationary inflection point. Consequently, the global optimum may also occur at the boundary of the feasible domain (e. g., at saturation or hitting $D_{\max}$).

(b) *Efficiency peak.* Noting that $\mathrm{ROI}_P(C) = S_P(C) - 1$, interior extrema of the ROI satisfy

$$\partial_C \; ROI_P(C)\big|_{C=C^\star} = 0 \iff MU_P(C^\star) = S_P(C^\star),$$

i. e., marginal utility equals average scalability. As above, without concavity, such points need not be unique nor correspond to a global maximum.

(c) *Budget frontier.* For objective (c), monotonicity of D_P implies that the maximum attainable damage under budget B is given by $D_P(B)$.

(d) *Damage frontier.* For objective (d), the optimal solution corresponds to the minimal cost at which the damage threshold $D_{\min}$ is reached; uniqueness holds if and only if $D_P(C)$ is strictly increasing in a neighborhood of that level.

Implications of non-concavity of the cost–damage function. Overall, the absence of concavity in $D_P(C)$ is a central feature of scalable attacks: It allows for multiple local optima, boundary solutions, and qualitative changes in optimal behavior as budget or execution scale varies. While our model explicitly permits non-concave damage functions, we note that non-concavity already stems from allowing the costs of attacks to incorporate economies of scale, refining classical economical analysis assuming diminishing returns [11].

5.2 Path Switching Under Scale

The attacker may formulate a ranking of favorable attacks for a single execution due to a cost–damage analysis. This ranking can change based on the budget, as observed in Fig. 5, indicated by the intersection of paths. However, switching paths *within an attack*, i. e., combining their most favorable parts, can be better than following one path only, e. g. for attacker goal (a), allowing path switching trivially cannot decrease the net profit since it also comprises sticking to a path.

Following that argumentation, we immediately observe that goals (c) and (d) can also be fulfilled better if path switching is permitted. Further, for goal (b) switching paths can—counter-intuitively—result in a *better* ROI, as the mediant inequality does not hold here. The intuition behind that is that a threshold damage can be surpassed if several attacks of bad individual scaling are combined. An example of beneficial path switching can be found in Sect. 5.3 below.

Allowing path switching turns the optimization into a resource-allocation problem: The optimal achievable damage as a function of budget is given by the supremal convolution of the per-path cost–damage functions, i. e., by optimally splitting the total cost across paths and summing their contributions[2]—a well-known setting in mathematical economics [8,3]. In a discretized model (integer number of executions), each option $\big(P, C(n)\big)$ constitutes an item; for any fixed P these options are mutually exclusive (selecting n executions excludes any other execution of that path), which can be encoded as a conflict-constrained knapsack formulation, see [22].

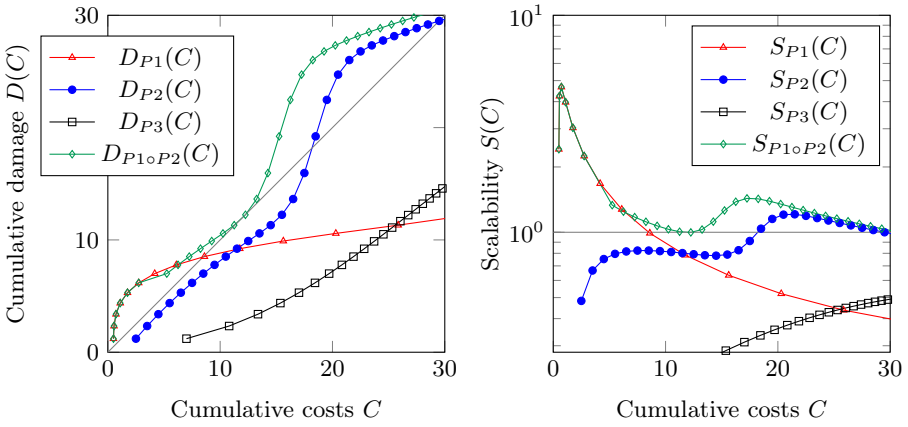
In this paper we do not propose a full algorithmic solver for switching under shared-node nonlinearities; instead, we formalize switching as a resource-allocation problem and use it to demonstrate that scalability can create incentives for mixed strategies that can dominate any single path. Designing efficient heuristics and tool support is important future work.

5.3 Path Reversal and Path Switching Example

Fig. 5 shows that multiple attack executions can induce path reversal beyond a certain budget marked by intersections of the graphs. Moreover, we note that scalability analysis fundamentally requires a *comparison of curves*: Although path $P1$ exhibits high scalability at small costs—three executions of $P1$ are optimal for attacker goal (b)—it does not "scale well" (in intuitive understanding) and does not simultaneously maximize attacker goal (a). As illustrated in Fig. 7a, net profit corresponds to vertical distances to the bisectrix, where path $P2$ attains a higher maximum than $P1$, highlighting that attacker goals may not align and require careful attacker characterization.

Now we consider *path switching*, i. e., combining different paths to achieve maximal damage: While attack $P1$ offers an attractive behavior at its beginning, its decreasing scalability makes this attack path less appealing for more

[2] Allowing path switching also requires keeping track of the attack executions for every node shared by several attack paths, because cost and damage of that node are in general nonlinear.



(a) Cost–damage plot with path switch.

(b) Scalability plot.

Fig. 7: Cost–damage plot (left) and scalability plot (right) for all paths. We note that path switching can substantially increase net profit (maximal vertical distance to bisectrix) and the scalability curve.

attacks. In contrast, $P2$ offers a good behavior at more executions. We hence combine these paths such that for the first six executions we follow path $P1$ and afterwards follow path $P2$[3]. We take care of the initial investment which switching to $P2$ requires and make sure not to double count the executions at their path intersection nodes. The results for the cost–damage plot and the scalability plot are shown in Figs. 7a and 7b respectively.

The corresponding path dominates all single paths in terms of cost–damage plot and scalability plot as depicted in Fig. 7 (except for the switching point). This attack hence yields greater damage at given costs and shows a better scalability than any single attack path. Consequently, extending the analysis from single attacks to multiple executions must also consider path switching, since it can reveal powerful attacks overlooked by standard risk assessment.

To assess how sensitive the observed path rankings and intersections are to moderate parameter uncertainty in this illustrative tree, we provide a sensitivity analysis in App. B.

6 Discussion

The central qualitative effect enabled by our model is that attacker preferences can *change with scale*. Because we lift costs to execution-indexed cumulative functions and analyze their budget-indexed view, the ranking of paths under common objectives (net profit, ROI, budget-/target-constrained) may reverse

[3] We note that finding the optimal switching point is in general a combinatorial problem, see Sect. 5.2.

as the number of executions grows. This helps explain why an attack vector that appears unattractive in a one-shot assessment may become preferable when amortized over many executions, and conversely why initially cheap vectors may lose their appeal once their marginal efficiency deteriorates.

A second key implication is that cost–damage relations need not exhibit globally diminishing returns. Many classical cost–benefit analyses rely (implicitly or explicitly) on regularity assumptions (e. g., concavity of a benefit function or unimodality of net benefit) to obtain a unique interior optimum from first-order conditions. In contrast, scalable attacks can naturally yield *non-concave* $D_P(C)$ (and hence non-concave net benefit and ROI) due to sublinear cost growth combined with nonlinear damage.

Limitations

Interdependencies. We treat paths independently and repetition as re-executing one chosen path. In reality, executions can change other paths' feasibility/costs (reuse, exposure, patching, burned access). Additionally, the system dynamics may be heavily altered by defender's reactions. This requires a state-dependent, sequential model; greedy re-selection need not be globally optimal.
Success ratio. We implicitly assume a static 100% success ratio if the required basic attacks are activated. Probabilistic success can be incorporated by treating the number of successful executions as a random variable $\mathcal{N}$ and replacing $D(n)$ by $\mathbb{E}[D(\mathcal{N})]$; we leave this extension to future work to keep the present contribution focused on compositional propagation and objective-induced reversals.
Solutions to the path-switching problem. While the specialized cost–damage optimization problem (only single executions of each attack and damages independent of the amount of repeated executions) is already NP-hard and has been proven to not be applicable to knapsack heuristics [20], new heuristic approaches for the discrete path-switching optimization can further foster the adaptation of execution-dependent costs and damages.
Limited application domain of economic models. While it is clear that the range of a damage function is limited to the total value of the system, the applicability of economic cost models is also limited. Attacks becoming infinitely expensive at an infinite number of executions is a simplification of reality. Once an asset is destroyed, there are no more attack executions to be expected.

7 Conclusion and Outlook

We have introduced scalability-aware attack tree analysis by lifting leaf costs to execution-indexed cumulative functions and propagating them compositionally through AND/OR connections to obtain path-level cost profiles and budget-indexed views. This exposes a core phenomenon that is hidden in static cost–damage analyses: Under repetition, cost–damage relations can become non-concave and attacker-optimal choices can reverse with scale. As a result, a path

that is optimal for a single execution may become suboptimal over many executions, and switching paths across executions can be economically optimal.

We illustrated these effects on representative attack-tree examples, demonstrating that conclusions drawn from single-execution or static cost–damage views may no longer hold once repetition is considered. While parameter calibration is inherently system-specific, the sensitivity experiments in App. B indicate that the qualitative phenomena observed in our running example persist under moderate estimation uncertainty. Although the extension represents a huge refinement of existing schemes, at the same time, several important aspects remain open, most notably implementable heuristics (and tool support) for computing beneficial path-switching strategies across executions.

Acknowledgments. This work was funded by the Topic Engineering Secure Systems of the Helmholtz Association (HGF) and supported by KASTEL Security Research Labs (structure 46.23.02).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. André, É., Lime, D., Ramparison, M., Stoelinga, M.: Parametric analyses of attack-fault trees. Fundamenta Informaticae **182**(1), 69–94 (2021)
2. Aslanyan, Z., Nielson, F.: Pareto efficient solutions of attack-defence trees. In: International Conference on Principles of Security and Trust. pp. 95–114. Springer (2015)
3. Bayón, L., García-Nieto, P., García-Rubio, R., Grau, J.M., Ruiz, M., Suárez, P.: The operation of infimal/supremal convolution in mathematical economics. International Journal of Computer Mathematics **93**(5), 735–748 (2016)
4. Bistarelli, S., Fioravanti, F., Peretti, P.: Defense trees for economic evaluation of security investments. In: First International Conference on Availability, Reliability and Security (ARES'06). pp. 416–423. IEEE (2006)
5. Brooks, F.P.: The mythical man-month. Datamation **20**(12), 44–52 (1974)
6. Buldas, A., Laud, P., Priisalu, J., Saarepera, M., Willemson, J.: Rational choice of security measures via multi-parameter attack trees. In: International Workshop on Critical Information Infrastructures Security. pp. 235–248. Springer (2006)
7. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Catastrophic cascade of failures in interdependent networks. Nature **464**(7291), 1025–1028 (2010)
8. Burgert, C., Rüschendorf, L.: On the optimal risk allocation problem. Statistics and Decisions-International Journal Stochastic Methods and Models **24**(1), 153–172 (2006)
9. DESTATIS: Auf einen Hausarzt oder eine Hausärztin kommen im Schnitt 1 264 Einwohnerinnen und Einwohner (Sep 2025), `https://www.destatis.de/DE/Presse/Pressemitteilungen/2025/08/PD25_N046_231.html`, accessed: March 30, 2026
10. Fraunhofer SIT: Abschlussbericht Sicherheitsanalyse des Gesamtsystems ePA für alle (Oct 2024), `https://www.gematik.de/media/gematik/Medien/ePA_fuer_alle/Abschlussbericht_Sicherheitsanalyse_ePA_fuer_alle_Frauenhofer_SIT.pdf`, accessed: December 11, 2025

11. Gordon, L.A., Loeb, M.P.: The economics of information security investment. ACM Transactions on Information and System Security (TISSEC) **5**(4), 438–457 (2002)
12. Hax, A.C., Majluf, N.S.: Competitive cost dynamics: the experience curve. Interfaces **12**(5), 50–61 (1982)
13. Hetzel, E., Nemes, M., Müller-Quade, J.: Attack Once, Compromise All? On the Scalability of Attacks. In: International Joint Conference on Electronic Voting. pp. 90–106. Springer (2025)
14. Iliaev, D., Oren, S., Segev, E.: A tullock-contest-based approach for cyber security investments. Annals of Operations Research **320**(1), 61–84 (2023)
15. Jürgenson, A., Willemson, J.: Serial model for attack tree computations. In: International Conference on Information Security and Cryptology. pp. 118–128. Springer (2009)
16. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 137–146 (2003)
17. Kordy, B., Kordy, P., Mauw, S., Schweitzer, P.: Adtool: security analysis with attack–defense trees. In: International conference on quantitative evaluation of systems. pp. 173–176. Springer (2013)
18. Kumar, R., Stoelinga, M.: Quantitative security and safety analysis with attack-fault trees. In: 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE). pp. 25–32. IEEE (2017)
19. Lopuhaä-Zwakenberg, M., Budde, C.E., Stoelinga, M.: Efficient and generic algorithms for quantitative attack tree analysis. IEEE Transactions on Dependable and Secure Computing **20**(5), 4169–4187 (2022)
20. Lopuhaä-Zwakenberg, M., Stoelinga, M.: Cost-damage analysis of attack trees. In: 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 545–558. IEEE (2023)
21. Mossel, E., Roch, S.: Submodularity of influence in social networks: From local to global. SIAM Journal on Computing **39**(6), 2176–2188 (2010)
22. Pferschy, U., Schauer, J.: The knapsack problem with conflict graphs. Journal of Graph Algorithms and Applications **13**(2), 233–249 (2009)
23. Rana, A., Gupta, S., Gupta, B.: A comprehensive framework for quantitative risk assessment of organizational networks using fair-modified attack trees. Frontiers in Computer Science **6**, 1304288 (2024)
24. Schäfer, B., Witthaut, D., Timme, M., Latora, V.: Dynamically induced cascading failures in power grids. Nature communications **9**(1), 1975 (2018)
25. Schneier, B.: Attack trees. Dr. Dobb's journal **24**(12), 21–29 (1999)
26. Shapiro, C., Varian, H.R.: Information rules: A strategic guide to the network economy. Harvard Business Press (1999)
27. Tullock, G.: Efficient rent seeking. Toward a theory of the rent-seeking society **97**, 112 (1980)
28. Verhulst, P.F.: Recherches mathématiques sur la loi d'accroissement de la population, vol. 18. Académie Royale de Bruxelles (1844)
29. Wright, T.P.: Factors affecting the cost of airplanes. Journal of the aeronautical sciences **3**(4), 122–128 (1936)

A Cost and Damage Functions

This appendix takes a closer look at laws from economics and how they can relate to attacks. While cost–damage curves are mostly assumed to be concave in traditional cybersecurity economics [11], we extend this by considering non-concave and convex functions for cost and damage individually. The laws selected below also stem from economics and capture different aspects of scaling behavior. It hence seems natural to transfer their validity to cost and damage functions. This selection does not claim completeness; in particular the framework is open to any (well-behaved) function shapes.

To determine cost and damage behavior under scale (choosing the parameters of the function), analysis of previous attacks, pre-existing knowledge (e. g., system stability thresholds), or thorough estimation has to be performed – common for practitioners. The cost and damage functions revisited here are summarized in Table 1 in Sect. 3.

A.1 Cost

First, we consider four ways costs can scale. The goal of this section is not to present a complete list, but to show the variety of different cost functions.

Sublinear Scaling (Wright's Law) The learning curve effect, derived from Wright's law [29], states that in industry, every time the number of produced goods is doubled, the costs are reduced by a fixed percentage. This effect is justified by the increase in experience and efficiency gained with every item that is produced. The experience curve effect, which is similar to the learning curve effect, has been identified in multiple industries with varying cost reduction percentages [12]. According to Wright's law, the unit cost c_i of the nth unit can be described as

$$c_n = c_1 \cdot n^{\log_2(b)},$$

where c_1 is the cost of the first unit and $1-b$ is the proportion by which the cost is reduced with every doubling of the goods produced, which can range from 10% to 25% according to [12]. The cumulative cost function is therefore

$$\bar{C}(n) = \sum_{i=1}^{n} c_i = \sum_{i=1}^{n} c_1 \cdot i^{\log_2(b)}, \tag{10}$$

or if the cost function is assumed to be continuous as discussed in Sect. 3.2, we obtain

$$C(n) = c_1 + \int_1^n c_1 \cdot n'^{\log_2(b)} \, \mathrm{d}n' = c_1 \cdot \big(1 + \frac{1}{1+\log_2(b)} \ (n^{1+\log_2(b)} - 1)\big), \tag{11}$$

with $b \in (0, 0.5) \cup (0.5, 1]$. For $b = 0.5$ we have the following:

$$C(n) = c_1\big(1 + \ln(n)\big). \tag{12}$$

Translating the learning curve effect to the scalability of attacks, produced goods become attack executions. As the attacker attacks their targets, they become more efficient and can reuse equipment, leading to less additional costs with every attack execution. Figure 8a shows how the total attacker costs grow sublinearly with every attack execution for $c_1 = 1$ and $b = 0.8$.

Superlinear Scaling (Brooks's Law) Next to such economies of scale, diseconomies of scale are possible as well. One example is the growth of communication costs and an increased effort of training new employees when a company expands, which is described as Brooks's law. In his book called "The Mythical Man-Month", Brooks explains that there are $\frac{n^2-n}{2}$ communication interfaces between n employees working on the same project, leading to a delay in finishing the task [5].

While Brooks's law mostly applies to companies with larger numbers of employees, an attacker could also suffer from diseconomies of scale. Some examples could be bottlenecks in computation power or equipment which is hard to acquire and cannot be reused for multiple attack executions. Aside from monetary costs, the risk of being caught might, e. g., increase with every attack execution, resulting in another diseconomy of scale. Figure 8b shows a quadratic cost increase with every attack execution.

Zeroth Order (Non-Rival Goods) The theory of non-rival costs describes goods which can be used by multiple parties without impeding others [26]. This surely applies to goods which can be replicated basically for free such as computer programs. The cumulative costs of those are close to constant with small additional costs for additional users:

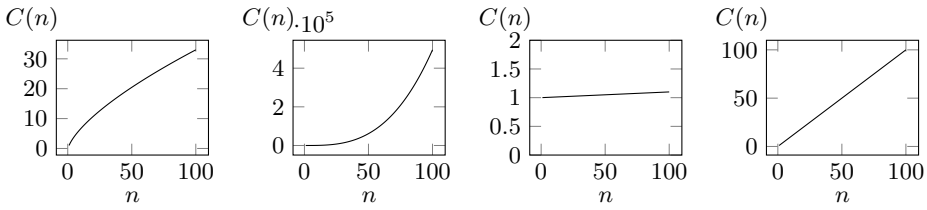
$$C(n) = \varepsilon \cdot n + b, \text{with } \varepsilon \text{ very small.} \tag{13}$$

The cumulative cost function is displayed for $\varepsilon = 0.001$ and $b = 1$ in Fig. 8c.

First Order (Constant Cost) The standard cost-behavior of multiple attacks assumes that the number of already executed attacks does not have any effect on the cost of another attack. This stems from repeated attack expenditures being independent of each other. Accordingly, examples can be found in the physical world, such as breaking into several houses which requires linearly more attack hours, getaway cars etc. The cumulative costs of such a class of attacks reads

$$C(n) = c_1 \cdot n, \tag{14}$$

where c_1 represents the cost for a single attack. Figure 8d shows the constant cumulative cost function for $c_1 = 1$.



(a) Sublinear. (b) Superlinear. (c) Zeroth order. (d) First order.

Fig. 8: Plots of cumulative attacker costs $C(n)$ with every attack execution n to illustrate different ways in which costs can scale.

A.2 Damage

Similarly to cost, damage can also exhibit nonlinear behavior.

Saturating Damage (Submodular) In case of a saturation of the damage due to a diminishing marginal impact, the damage function can be described with a saturation curve in the form of a Tullock contest success function [27]:

$$D(n) = D_{\max} \cdot \frac{n^b}{n^b + a}, \tag{15}$$

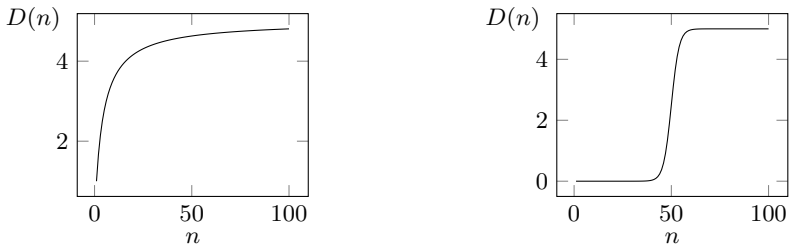
where $D_{\max}$ is the total value of the system under attack and $\frac{D_{\max}}{1+a}$ is the impact of the first attack execution. b describes the ratio of the impact of an attack and the damage that is possible. This can be interpreted as a Tullock contest where the two players are the attacker and the defender [14].

In such a system, large parts are destroyed with the first few attack executions, but then the cumulative damage converges to a saturation point. Fig. 9a shows the damage curve for $D_{\max} = 5$, $a = 4$, and $b = 1$.

Cascading Damage (S-shaped) In interdependent infrastructures, repeated stress can trigger superlinear escalation. While the first few attacks might be covered by redundancies, further attacks lead to a sudden destruction of the system. Once most of the system is destroyed, further attacks again yield only small additional damage, since the value of the system is limited. This behavior can be captured by a logistic function borrowed from population growth models [28]:

$$D(n) = \frac{D_{\max}}{1 + \exp\left(-b \cdot (n - n_0)\right)}, \tag{16}$$

where $D_{\max}$ is the total value of the system under attack, b is the logistic growth rate and n_0 is the number of executions necessary to damage half of the system. The damage curve in Fig. 9b is plotted for $D_{\max} = 5$, $b = 0.1$, and $n_0 = 50$.



(a) Saturating. (b) Cascading.

Fig. 9: Plots of cumulative damage $D(n)$ with every attack execution n to illustrate different ways in which damage can scale.

B Sensitivity and Regime Analysis (Illustrative Tree)

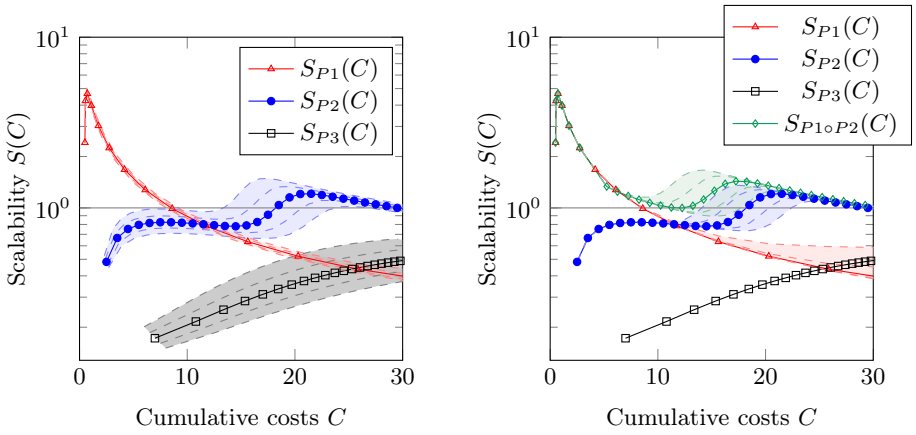
This appendix complements the illustrative example (Sects. 4.1 and 5.3) by examining how sensitive the observed scalability behavior is to parameter choices. We distinguish between local sensitivity analysis, modeling moderate estimation uncertainty in cost and damage parameters, and regime analysis, exploring how qualitative behavior changes when structural parameters governing scaling are varied. The goal is not to generalize beyond the example tree, but to clarify which conclusions are robust to small perturbations and which ones depend on specific modeling regimes.

B.1 Sensitivity Analysis

Setup. We study robustness of the qualitative phenomena in the illustrative tree under moderate parameter uncertainty. For each path P_i, we perturb its cumulative leaf-cost instantiation by $\pm 20\%$ while keeping the offset (zeroth order) constant. We recompute the propagated $C_{P_i}(n)$ and accordingly $n_{P_i}(C)$, and obtain $S_{P_i}(C)$ (Def. 4). We deem the scalability behavior robust if (i) the qualitative curve regimes (sub-/linear/superlinear) remain, and (ii) the ordering changes only by shifts of intersection points, but without eliminating the observed path reversal/ switching incentive. The results are shown in Fig. 10a.

Additionally, we change the threshold parameter n_0 of the logistic damage vertex v_{10} (see Sect. 4) by $\pm 20\%$ and recompute the scalability. We deem the path switching decision robust if (i) a domination of a path is not changed and (ii) the path switching only shifts with respect to switching the execution number. The results are shown in Fig. 10b.

Evaluation. We observe in Fig. 10a a path-specific sensitivity induced by different cost-growth regimes (e. g., linear vs. nonlinear cumulative costs), while the qualitative ordering and curve shapes remain unchanged in this illustrative tree. For Fig. 10b we note that since P_1 and P_2 share the same damage profile in the illustrative instance, their relative ranking is unchanged; however, delaying



(a) Robustness wrt. estimation uncertainty of costs.

(b) Robustness wrt. threshold parameter of damage vertex v_{10}.

Fig. 10: Sensitivity Analysis wrt. estimation uncertainties. The resulting thin bands indicate a change of ±10% or ±20%.

damage onset lowers attainable peak scalability because more budget must be invested before comparable damage materializes.

Across all perturbations, the qualitative shape of the cost-indexed scalability curves and the resulting path ranking remain unchanged in our illustrative tree, indicating local robustness of the observed preference patterns. The uncertainty bands differ across paths, which is expected because the paths exhibit different cost-growth regimes (e. g., linear vs. sub-/superlinear), so identical relative perturbations translate into different absolute deviations over budget. Additionally, path switching is still preferential with respect to a single path execution.

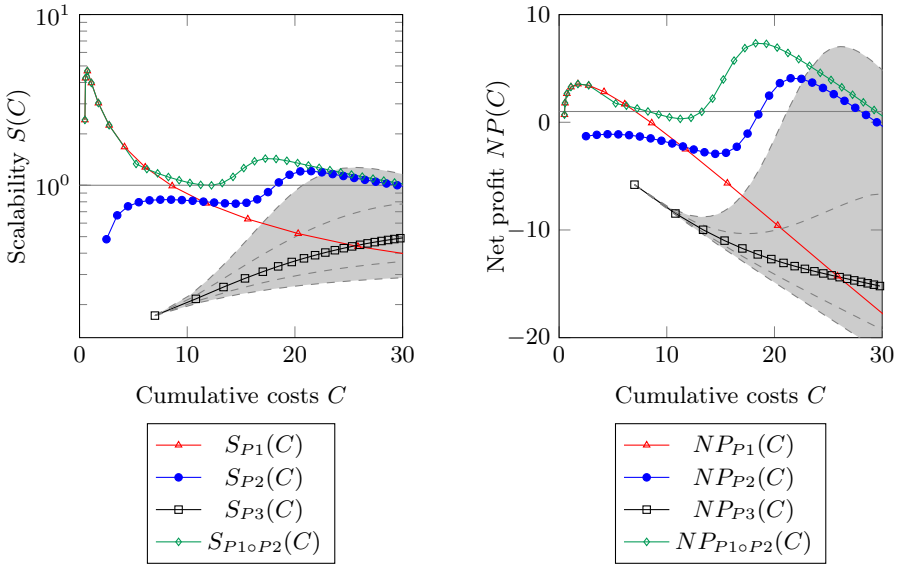
Overall, this experiment supports that the example's conclusions are not the result of fine-tuning a single set of cost parameters.

B.2 Regime Analysis

Setup. Varying the Wright-law learning parameter b (App. A) constitutes a regime change in the strength of economies of scale. We therefore separately sweep the parameter b over an extended range of $[0.45, 0.75]$ and report how profitability and ranking change, to distinguish structural phenomena from parameter fine-tuning. We note that this change in learning rate is substantial. The resulting scalability plot is shown in Fig. 11a, while Fig. 11b showcases the corresponding net profit deviation.

Evaluation. The Wright-parameter b variation of ±25% changes the strength of sublinear cost growth and therefore constitutes a regime change rather than a small estimation error. Under lower b (stronger learning effects), $P3$ becomes profitable and its net-profit curve improves, surpassing $P1$ in part of the bud-

get range; however, it remains less profitable than the switching strategy that combines the favorable early behavior of $P1$ with the superior scaling of $P2$. This demonstrates that economies-of-scale parameters can qualitatively change profitability regimes, and switching can remain dominant even when alternative paths become individually profitable, reinforcing the relevance of switching-aware analysis.



(a) Scalability plot wrt. to changes in the learning parameter.

(b) Net profit plot wrt. to changes in the learning parameter.

Fig. 11: Regime analysis wrt. changes in the Wright learning parameter. The thin dashed lines represent $\pm 12.5\%, \pm 25\%$ changes of b.

B.3 Take-away

These sweeps are not intended to generalize quantitatively beyond the illustrative tree; rather, they provide (i) a local robustness check under moderate parameter uncertainty and (ii) a regime analysis showing which parameters can induce qualitative shifts (profitability and path preference). Together, they support the methodological claim that scalability-aware attack-tree analysis can reveal preference reversals and switching incentives that are not visible in static, single-execution assessments.

C Electronic Patient Record Example Continued (Section 4.2)

The security analysis of the German electronic patient record ePA specifies success probabilities instead of costs for the leaf nodes of the attack trees. The success probability is multiplied with the probability of occurrence, which is assumed to be 1 if an attacker is in principle able to perform the attack, and 0 otherwise [10]. Since the report does not provide any cost estimates, we assume that the first executions of two attacks are equally expensive if they have the same success probabilities.

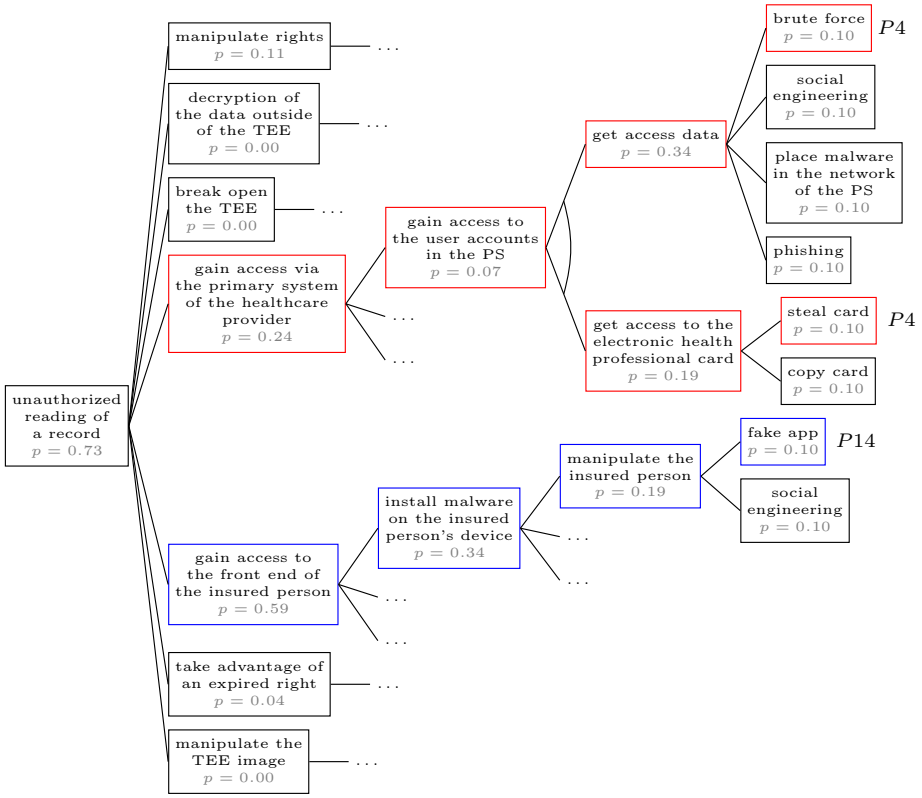


Fig. 12: Attack tree to analyze the risk of an attacker reading a patient record without authorization, translated from [10]. The AND connection is marked with an arc. Access via the front end of the insured person seems to have the highest risk due to the success rate at first, but multiple patient records can be accessed if the attacker chooses the path via the healthcare provider, leading to a better attack scalability.

The ePA study is used purely as an illustrative demonstration of how repeated-execution effects can change rankings; the cost instantiations derived from success probability categories are not claimed to be calibrated estimates. The qualitative phenomenon we highlight is the scalability gap between "single-record" and "many-records-per-compromise" vectors under repetition.

Let us look at the upper attack paths shown in Fig. 12 in detail. To find the minimal cut sets, we observe that the attacker has to combine one of the four elementary attacks to get the healthcare provider's access data with one of the two elementary attacks to get the electronic health professional card of the same healthcare provider, which are used in combination to authenticate the healthcare provider. This results in eight different paths in this part of the tree.

In a second step, we assign cumulative cost functions to the leaf nodes. Many elementary attacks such as brute force attacks or stealing a physical card have first order cost functions, since no economies of scale can be taken advantage of. Other attacks, such as the development of malicious software or fake apps, have high initial costs, but can be reused for multiple attacks with almost no additional costs. Superlinear or sublinear cost functions are less common in this example, except for small learning effects that the attacker can benefit from. However, for the quantification of these learning effects, data about previous attacks would have to be gathered, which is why they were neglected in this analysis.

The attacker could, e. g., choose to steal the electronic health professional card and use a brute force attack to learn the corresponding access data. This results in $C_{P4}(1) = \tilde{c} + \tilde{c} = 2 \cdot \tilde{c}$, since the elementary attacks are equally likely and are connected via an AND node. However, to read a second record, the attacker can again use the same card and access data, since one healthcare provider has access to the data of all of their patients. Assuming that every general practitioner and every insured person uses the ePA, one practice would have access to the data of approximately 1300 patients [9]. Only once the attacker has successfully read all of the patient records accessible through one healthcare provider, they have to invest again. At this point, the brute force attack and the theft of another card are as expensive as at the first attack execution. This leads to the following step function for the cumulative costs of the path[4]:

$$C_{P4}(n) = \begin{cases} 2 \cdot \tilde{c}\ 1 \leq n \leq 1300 \\ 4 \cdot \tilde{c}\ 1300 < n \leq 2600 \\ 6 \cdot \tilde{c}\ 2600 < n \leq 3900 \cdot \\ \vdots \quad \vdots \end{cases}$$

The goal of this section is not to provide a complete security analysis of the ePA, but to demonstrate our framework. For a simpler presentation, we therefore omit elementary attacks with a success probability of 0 or paths that include an AND connection with such a node. This leads to a total number of 62 minimal cut sets and therefore 62 ways an external attacker could try to read a record without authorization. 10 out of these 62 paths are via the "gain access via

[4] To make this cost function invertible, a term of $\varepsilon \cdot n$ would have to be added to each segment.

the primary system of the healthcare provider" node and 27 paths are via the "gain access to the front end of the insured person" node. The cumulative cost functions of the elementary attacks of these 37 paths are summarized in Table 2.

Table 2: Cost functions of the elementary attacks in the ePA example, divided into attacks on the insured person on the top and attacks on the healthcare provider on the bottom of the table

Type	Cost Function	Elementary Attacks
Zeroth order	$C(n) = \tilde{c} + 0.1 \cdot n$	Fake app, vulnerability in the updater, supply chain attack on legitimate software
First order	$C(n) = \tilde{c} \cdot n$	Social engineering, click-/tabjacking within an existing user session, physical access to user session, low entropy of session ID, session fixation, identify the front end user, find out the phone number of the user, call the user, send a phishing email, phishing via conventional mail, steal the smartphone, steal the device identifier, steal the device token, steal the recovery key, steal the email access data, steal the electronic health card, copy the card
First order	$C(n) = 2\tilde{c} \cdot n$	Brute force
Step function	$C(n) = \begin{cases} \tilde{c} & 1 \leq n \leq 1300 \\ 2 \cdot \tilde{c} & 1300 < n \leq 2600 \\ 3 \cdot \tilde{c} & 2600 < n \leq 3900 \\ \vdots & \vdots \end{cases}$	Brute force, social engineering, malware in the network of the healthcare provider, phishing, steal electronic health professional card, copy card
Step function	$C(n) = \begin{cases} \tilde{c} & 1 \leq n \leq 1300 \\ 1.1 \cdot \tilde{c} & 1300 < n \leq 2600 \\ 1.2 \cdot \tilde{c} & 2600 < n \leq 3900 \\ \vdots & \vdots \end{cases}$	Zero-day vulnerability, other known vulnerabilities

Figure 13 shows the cumulative cost functions of the 37 selected attack paths identified via the minimal cut sets. The cost functions of the paths were computed with the help of the cost functions of the elementary attacks according to Eq. 1. Paths with the same cumulative costs are displayed as just one curve. The

red and orange solid curves of paths $P4 - P13$ belong to attack vectors via the healthcare provider, while the blue and green dashed curves of paths $P14 - P40$ correspond to attacks via the insured person.

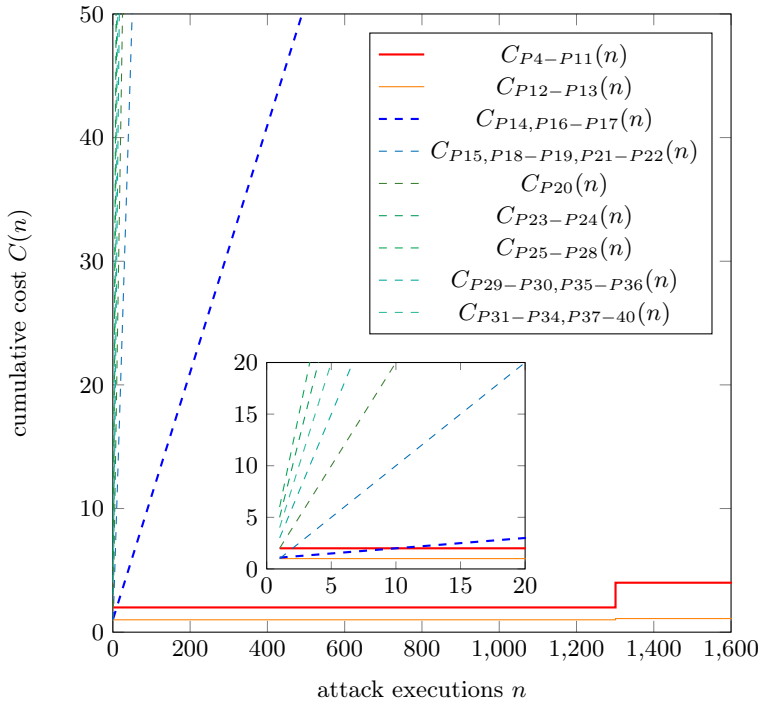
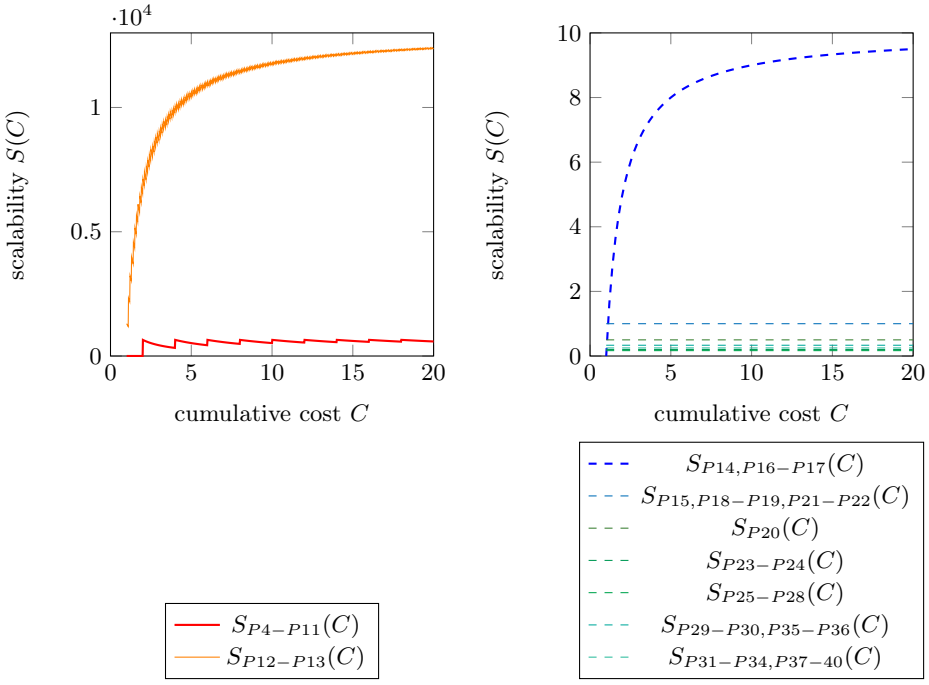


Fig. 13: Cumulative cost curves of the attack tree on unauthorized reading of a record with $\tilde{c} = 1$. Paths depicted in Fig. 12 are drawn thicker.

Even after addition, the different types of cost function can be clearly identified in the plot. The costs of attacks with a first order cost function rise sharply with the number of attack executions (light blue and green dotted lines). The dark blue dotted line contains elementary attacks with a zeroth order cost function, which can be repeated with less additional costs per execution. The step functions (red and orange solid lines) correspond to very cost-efficient attacks.

Although the medical data of, e. g., public figures like politicians and celebrities might be a higher value target for an attacker than other persons' patient records, we consider the impact of every path to be $D(n) = d_1 \cdot n$, i. e., the same for every record the attacker manages to read. For simplicity of presentation, we further assume $d_1 = 1$ and that there is no additional damage at internal nodes. Such additional damage could, e. g., be the replacement costs of a patient's electronic health card or smartphone if it was stolen in the attack.

After a variable transformation according to Eq. 3, the cumulative cost and damage functions can be plugged into Eq. 4 to find the scalability plots according to Def. 2. Figures 14a and 14b show the cost-indexed scalability ratio plots of the attack paths via the healthcare provider and the insured person, respectively. Please note the difference in y-axis scaling.



(a) Attack vectors via the healthcare provider.

(b) Attack vectors via the insured person.

Fig. 14: Selected cost-indexed scalability ratio curves of the attack tree on unauthorized reading of a record with $\tilde{c} = 1$ and $D(n) = n$.

All scalability curves in Fig. 14b approach a limit and do not exhibit any local extrema, as for many executions the cost–damage ratio approaches the ratio of their highest order terms (linear terms in this case). An attacker with an unlimited budget can always invest more without obtaining a worse cost–damage ratio of any of the attack paths. However, for all except the top two curves in Fig. 14b, the scalability stays below a value of 1, which means that the costs are higher than the impact. Having a closer look at the scalability curves in Fig. 14a, an underlying sawtooth-like shape can be observed. This is due to the steps in the cumulative cost functions. Only if enough cost units are invested, the attacker can reach the next impact threshold. Until then, the investment is in vain and this is shown in the short declining sections of the scalability curves.