Upper Bounds for Adversaries' Utility in Attack Trees

Ahto Buldas[1,2,3,⋆] and Roman Stepanenko[2]

[1] Cybernetica AS
[2] Tallinn University of Technology
[3] Guardtime AS

Abstract. Attack trees model the decision making process of an adversary who plans to attack a certain system. Attack-trees help to visualize possible attacks as Boolean combinations of atomic attacks and to compute attack-related parameters such as cost, success probability and likelihood. The known methods of estimating adversarie's utility are of high complexity and set many unnatural restrictions on adversaries' behavior. Hence, their estimations are incorrect—even if the computed utility is negative, there may still exist beneficial ways of attacking the system. For avoiding unnatural restrictions, we study *fully adaptive adversaries* that are allowed to try atomic attacks in arbitrary order, depending on the results of the previous trials. At the same time, we want the algorithms to be efficient. To achieve both goals, we do not try to measure the exact utility of adversaries but only *upper bounds*. If adversaries' utility has a negative upper bound, it is safe to conclude that there are no beneficial ways of attacking the system, assuming that all reasonable atomic attacks are captured by the attack tree.

1 Introduction

We live in the world where information is extremely valuable. Many of our activities depend on access to information which is correct and up to date. Even minor discrepancies in such things as on-line traffic schedules can cause huge inconveniences. It is crystal clear that information security is of utmost importance to governments and enterprises. Leakage of state secrets can cause conflicts between countries, and for commercial entities loss of their trade secrets may cost not only huge sums of money but also cause them to go bankrupt. Many security features have been introduced into modern information systems. It could be possible to talk about encryption, authentication and authorization schemes, various other technical solutions like firewalls, intrusion detection systems and so on. However even having introduced all those security measures it is difficult to give a quantitative answer how secure the information protected by them really is. There are many techniques of risk assessment available, however most of

⋆ This research has been supported by Estonian Science Foundation, grant No. 8124, and by the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science (EXCS).

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 98–117, 2012.
© Springer-Verlag Berlin Heidelberg 2012

them are not suitable for applying to information systems. For example, using statistical data to assess the likelihood of a threat may turn out to be impossible in practice—the very field is quite new and victims usually do not make security incidents public, which means that no statistical data is available. But this doesn't mean there is no hope in finding useful methods for information security.

Attack tree analysis, which is quite similar to fault tree analysis [6], is one of the promising methods. The idea behind attack trees is that multi-stage attacks against information systems can be decomposed into simple atomic attacks against the components of the system. Provided with the security metrics for the atomic attacks and a computational model it could be possible to estimate adversaries' expected utility which would allow us to talk about quantitative security of the system. Attack tree analysis has been used to analyze the Border Gateway Protocol [3], online banking systems [5], as well as large-scale e-voting systems [2].

There exists a handful of quantitative attack tree models, however they are plagued by important problems. The ones that appeared the earliest do not account for economical feasibility of attacks, while the more recent ones put unnatural restrictions on the actions available to the adversary. A good example of those restrictions is that some of the models require adversaries to fix the order of their (atomic) attacks in advance and do not allow adjustments of attacking strategies, while it is more natural to expect that the adversary chooses the next atomic attack adaptively, by taking into account the results of the previously tried atomic attacks. It is evident that such kind of model does not cover all attack possibilities and does not guarantee the absence of beneficial attacks against the system, even if all reasonable atomic attacks were taken into account in the model.

Only by being able to capture all reasonable ways of breaking the system, we could prove that the system is secure, and since the earlier models do not have this quality, a new approach to the problem is needed. Instead of computation-extensive methods for finding exact utilities of restricted adversaries, we have to find computationally lightweight methods for computing upper bounds of the utility of fully-adaptive adversaries. This way by showing that if the largest possible average utility of an economically oriented adversary is negative, we prove that the system is secure.

The aim of this paper is to introduce some of the available models for attack tree evaluation and to comment on their flaws as well as to present a new fully-adaptive model for computing upper bounds of the adversaries utility which is free of those problems. In Section 2, we outline the state of the art in the field of attack tree models, explain our motivation and sum up the main results. Section 3 outlines the main theoretical concepts of attack trees with fully adaptive adversaries. In Section 4, we present and analyze two composition rules that can be used in order to find efficiently computable upper bounds for the utility of fully adaptive adversaries. In Section 5, we describe another method that strengthens the adversary by assuming that every attack can be repeated arbitrary number of times. Some numerical examples are presented in Appendix A.

2 State of the Art, Motivation and Results

2.1 Attack Trees and Computational Models

Attack trees are models in which event algebra is used to visualize the decision making process of an adversary who decided to attack a certain system. In each step of the attack tree analysis, an attack $\mathcal{A}$ (as an event) is decomposed into several simpler attacks $\mathcal{A}_1, \ldots, \mathcal{A}_n$, that are defined to be the *child-attacks* of $\mathcal{A}$. In the visual description, $\mathcal{A}$ is represented as a node with $\mathcal{A}_1, \ldots, \mathcal{A}_n$ to be its child nodes. There are two types of decompositions used in the attack tree:

- AND-decomposition $\mathcal{A} = \mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n$ means that $\mathcal{A}$ happens (as an event) if and only if all child attacks $\mathcal{A}_1, \ldots, \mathcal{A}_n$ succeed.
- OR-decomposition $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ means that $\mathcal{A}$ happens if and only if at least one of the child attacks $\mathcal{A}_1, \ldots, \mathcal{A}_n$ succeed.

A tree-like structure (in general, a directed acyclic graph) is obtained when these two rules are used recursively several times to decompose $\mathcal{A}$ into simpler attacks. Attacks that are not decomposed in such a recursive process, are called *atomic attacks*. They correspond to the leaves of the attack-tree. To summarize, the attack tree analysis represent an attack $\mathcal{A}$ as a monotone Boolean formula $\mathcal{A} = \mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ of the atomic attacks $\mathcal{X}_1, \ldots, \mathcal{X}_m$.

Attack trees are useful not only for visualization, but also for computing several attack-related parameters such as cost, success probability, feasibility and likelihood, as shown by by Weiss [13] and Schneier [11]. Mauw and Oostdijk [9] presented general soundness rules for the computational semantics of attack-trees, which state that the semantics must be invariant under any transformation of the formula $\mathcal{A} = \mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ that does not change its Boolean function. Though already the early works used attack trees for many many different security-related parameters, they estimated just one of them at a time. Buldas et al. [1] presented a multi-parameter game-theoretic model to estimate the expected utility $U(\mathcal{A})$ of an adversary who tries to make $\mathcal{A}$ happen. Protecting a system against rational adversaries means that the security measures of the system should guarantee that $U(\mathcal{A}) \leq 0$ for all reasonable attacks $\mathcal{A}$.

To estimate $U(\mathcal{A})$, the model of [1] uses computational rules for AND and OR nodes to compute the game theoretic parameters of nodes based on the parameters of their child nodes. Their algorithm works in time linear in the number of nodes in the attack tree (i.e. the size of the Boolean formula $\mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$). Jürgenson and Willemson [8,7] showed that the computational semantics of the model does not satisfy the general requirements of Mauw and Oostdijk [9]. Jürgenson and Willemson proposed two new consistent models for computing exact utility of the adversary. In their so-called *parallel model* [8], the adversary tries to satisfy the Boolean function $\mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ by choosing a subset $\mathcal{S} \subseteq \{\mathcal{X}_1, \ldots, \mathcal{X}_m\}$ of atomic attacks and trying all of them independently in parallel. In [7], they refined their model by assuming that the atomic attacks (of the subset) are tried in certain (optimal) order σ and the adversary may skip the atomic attacks in the order in case they would not increase the probability

of materializing the root attack. They showed that the outcome $U_\sigma^{\mathrm{JW}}(\mathcal{A})$ of the adversary in their new (so-called *serial*) model always supersedes the outcome $U^{\mathrm{P}}(\mathcal{A})$ in the parallel model for any ordering σ, i.e. $U^{\mathrm{P}}(\mathcal{A}) \leq U_\sigma^{\mathrm{JW}}(\mathcal{A})$. They payed with the cost though, because while the parallel algorithm works in time $O(2^m)$ (number of terms in a DNF), the serial model uses time $O(m!)$.

Niitsoo [10] showed that the attack-skipping rule of the serial model of Jürgenson and Willemson [7] is not optimal and proposed a new rule inspired by standard *decision theory* by which an atomic attack is skipped if and only if this increases the expected outcome. Niitsoo showed that in his so-called *decision-theoretical model* the adversary's utility $U^{\mathrm{DT}}(\mathcal{A})$ is at least as large as in the serial model of [7], i.e. $U_\sigma^{\mathrm{JW}}(\mathcal{A}) \leq U_\sigma^{\mathrm{DT}}(\mathcal{A})$ for any order σ. He also showed that in case of a certain fixed natural order σ of the atomic attacks, the exact utility can be computed in time liner in the size of the attack tree.

2.2 Shortcomings of the Previous Computational Models

None of the three models [8,7,10] captures all possibilities of the adversary. In both the serial model [7] and the decision-theoretic model [10], the order σ of the atomic attacks is fixed and cannot be adjusted by the adversary during the attack. It is more logical for the adversary to choose the next attack based on the results of the previous trials. Obviously, $\max_\sigma U_\sigma^{\mathrm{DT}}(\mathcal{A}) \leq U^{\mathrm{FA}}(\mathcal{A})$, for the utility $U^{\mathrm{FA}}(\mathcal{A})$ of the adversary in such a *fully adaptive model*, but $U^{\mathrm{FA}}(\mathcal{A})$ was considered in [7] to be too complex to estimate. As the inequality may be strict, it might be that $\max_\sigma U_\sigma^{\mathrm{DT}}(\mathcal{A}) < 0$, but still $U^{\mathrm{FA}}(\mathcal{A}) > 0$, which means that negative utility upper bounds in terms of the serial and decision-theoretic models [7,10] do not guarantee that there are no beneficial adaptive attacks.

2.3 Our Motivation and Goals

The main goal of the attack tree analysis is to justify that a system is secure, assuming that the attack-tree captures all reasonable attacks. The computational models proposed so far are not quite suitable for such analysis because of unnatural restrictions on the behavior of adversaries. Instead of computation-extensive methods for finding exact utilities of restricted adversaries, we have to find *computationally lightweight* methods for computing *upper bounds* of the utility of *fully-adaptive* (or even artificially overpowered) adversaries.

2.4 Main Results of this Work

The starting point of this work is that even though the exact value of $U(\mathcal{A}) = U^{\mathrm{FA}}(\mathcal{A})$ is hard to compute, there might exist rough but easily computable upper bounds for $U(\mathcal{A})$. We first turn back to the method of AND- and OR-rules that was first proposed in [1] and study the following two natural *negativity rules*:

- AND-rule: If $U(\mathcal{A}_i) \leq 0$ for an $i \in \{1, \ldots, n\}$, then $U(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n) \leq 0$.
- OR-rule: If $U(\mathcal{A}_i) \leq 0$ for every $i \in \{1, \ldots, n\}$, then $U(\mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n) \leq 0$.

We prove that the AND-rule holds universally and that the OR-rule holds if $\mathcal{A}_1, \ldots, \mathcal{A}_n$ do not contain common atomic attacks. We show that the OR-rule does hot hold in the general case. The main reason is that if $\mathcal{A}_1$ and $\mathcal{A}_2$ contain a common atomic attack $\mathcal{X}$, then trying $\mathcal{X}$ contributes to both attacks $\mathcal{A}_1$ and $\mathcal{A}_2$ and there may exist attacking strategies for $\mathcal{A}_1 \vee \mathcal{A}_2$ that play $\mathcal{A}_1$ and $\mathcal{A}_2$ "in parallel" and has utility larger than $\max\{U(\mathcal{A}_1), U(\mathcal{A}_2)\}$.

To make the general OR-rule work in the general case, we introduce the so-called *cost reduction* technique, that uses the fact that the statement $U(\mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n) \leq 0$ will follow from somewhat stronger assumptions $U(\mathcal{A}'_1) \leq 0$, ... , $U(\mathcal{A}'_n) \leq 0$, where $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ are attacks in which the cost parameters of the atomic attacks $\mathcal{X}_1, \ldots, \mathcal{X}_m$ have been artificially lowered. For example, if $\mathcal{X}'_1$ is the same atomic attack as $\mathcal{X}_1$ the cost of which is half the original cost, then

$$U(\mathcal{X}_2 \wedge \mathcal{X}'_1) \leq 0 \text{ and } U(\mathcal{X}'_1 \wedge \mathcal{X}_3) \leq 0 \;\; \Rightarrow \;\; U((\mathcal{X}_2 \wedge \mathcal{X}_1) \vee (\mathcal{X}_1 \wedge \mathcal{X}_3)) \leq 0 \;\; .$$

We also show that there is an $O(m \log m)$ algorithm to determine the optimal attacking strategy in the case $\mathcal{A} = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_m$, where $\mathcal{X}_1, \ldots, \mathcal{X}_m$ are atomic. This is possible because there exists an easily computable invariant r (so-called *cost-nonsuccess ratio*), such that $\mathcal{X}_i$ must be tried before $\mathcal{X}_j$ if and only if $r(\mathcal{X}_i) < r(\mathcal{X}_j)$. The question of existence of such invariants was left open in [10] and hence we completely solved this open question.

Together with the cost-reduction technique this will give us the following method of determining upper bounds of $U(\mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m))$. First, represent the Boolean function $\mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ is a Disjunctive Normal Form (DNF), reduce the cost of those atomic attacks that belong to more than one term of the DNF, determine the utility of each term in $O(m \log m)$ time. Though, the number of terms in a DNF can be large, it is still much less than the number $m!$ of all orderings of the atomic attacks.

Finally, we generalize the concept of an adversary so that it is possible to retry some of the atomic attacks in the case of failure. We assume that each atomic attack is either not repeatable or can be repeated arbitrarily many times. We show that we can reduce this kind of adversaries to the case of ordinary non-repeatable model by just modifying the parameters of repeatable atomic attacks. In such a model, if the utility of an adversary is denoted by $U_\infty(\mathcal{A})$, then $U(\mathcal{A}) = U^{\mathrm{FA}}(\mathcal{A}) \leq U_\infty(\mathcal{A})$. Hence, if we prove that $U_\infty(\mathcal{A}) \leq 0$, this also implies $U(\mathcal{A}) \leq 0$. We also show (Theorem 8) that in the model where all attacks are repeatable, we can use the DNF-method without cost reduction, which means that though the adversary is only mildly strengthened, we are able to compute upper bounds in the *fully adaptive model* with approximately the same cost as that of computing $U^{\mathrm{P}}(\mathcal{A})$ in the parallel model of Jürgenson and Willemson [8].

3 Attack Trees with Fully Adaptive Adversaries

3.1 Notation

If $\mathcal{F}(x_1, \ldots, x_m)$ is a Boolean formula and $v \in \{0, 1\}$, then by $\mathcal{F}_{x_j=v}$ we mean a Boolean formula $\mathcal{F}'(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_m)$ derived from $\mathcal{F}$ by the

assignment $x_j := v$. By $\mathcal{F} \equiv 1$ we mean that $\mathcal{F}$ is identically true (i.e. is a tautology), and by $\mathcal{F} \equiv 0$, we mean that $\mathcal{F}$ is identically false. By a *min-term* of a Boolean formula $\mathcal{F}(x_1, \ldots, x_m)$, we mean a Boolean formula $M(x_1, \ldots, x_m)$ of type $x_{i_1} \wedge x_{i_2} \wedge \ldots \wedge x_{i_k}$ such that $M(x_1, \ldots, x_m) \Rightarrow \mathcal{F}(x_1, \ldots, x_m)$ is a tautology. We say that M is a *critical min-term* of M if non of the sub-terms $M'(x_1, \ldots, x_m) = x_{i_1} \wedge \ldots \wedge x_{i_{j-1}} \wedge x_{i_{j+1}} \wedge \ldots \wedge x_{i_k}$ is a min-term of M.

3.2 Attack Trees, Strategies and Utility

Definition 1 (Attack tree). *An attack tree $\mathcal{A}$ consists of the next components:*

- *A finite number of* atomic attacks $\mathcal{X}_1 \ldots \mathcal{X}_m$*, each attack $\mathcal{X}_i$ having the following parameters:* success probability p_i*,* failure probability q_i*,* (preparation) cost C_i *(a real number), and* penalty Π_i *(a real number).*
- *A negation-free (monotone) Boolean formula $\mathcal{F}(x_1, \ldots, x_m)$, where x_j are the input variables that correspond to the atomic attacks $\mathcal{X}_j$.*
- *The prize P (a non-negative real number).*

Definition 2 (Subtree). *By a* subtree $\mathcal{B}$ *of an attack tree $\mathcal{A}$ with Boolean formula $\mathcal{F}_{\mathcal{A}}(x_1, \ldots, x_m)$ we mean an attack tree with the same parameters as $\mathcal{A}$, except that the Boolean formula $\mathcal{F}_{\mathcal{B}}$ of $\mathcal{B}$ is in the form*

$$\mathcal{F}_{\mathcal{A}}(x_1, \ldots, x_m) = \mathcal{F}'(\mathcal{F}_{\mathcal{B}}(x_1, \ldots, x_m), x_1, \ldots, x_m) \ , \tag{1}$$

where $\mathcal{F}'$ is a negation-free Boolean formula $\mathcal{F}'$. Note that (1) is an identity between Boolean formulae*, not just between Boolean functions.*

Each attack tree $\mathcal{A}$ represents a *one-player game*, where the adversary (the player) can choose and execute atomic attacks one by one. At any stage of the game, the adversary is always allowed to *give up*, i.e. stop playing the game.

Definition 3 (Attack game). *By an* attack game *we mean a one-player game, every instance of which is an attack tree $\mathcal{A}$ (with Boolean formula $\mathcal{F}$), whereas*

- *if $\mathcal{F} \equiv 1$, the game is* won *and the adversary gets the prize P;*
- *if $\mathcal{F} \equiv 0$, the game is* lost *and the adversary does not get the prize.*

A game $\mathcal{A}$ that is neither won nor lost, the adversary may choose an atomic attack $\mathcal{X}_j$ after which it has to pay the costs C_j of $\mathcal{X}_j$ and the following happens:

- *with probability p_j, the game is reduced to $\mathcal{A}_{x_j=1}$ (with formula $\mathcal{F}_{x_j=1}$);*
- *with probability q_j, the game is reduced to $\mathcal{A}_{x_j=0}$ (with formula $\mathcal{F}_{x_j=0}$); and*
- *with probability $1 - p_j - q_j$, the adversary* gets caught*, i.e. it has to pay the penalty Π_j and the game is over. Formally, we denote this case by $x_j = \perp$.*

Definition 4 (Strategy). *A* strategy *S for an attack tree $\mathcal{A}$ is a rule that for any sequence of assignments $\langle x_{j_1} = v_1, \ldots, x_{j_k} = v_k \rangle$ (where $v_j \in \{0, 1\}$) that represent the previous moves, and possibly some auxiliary information, either points to the next atomic attack $\mathcal{X}_{j_{k+1}}$ to try, or decides to give up the game.*

Definition 5 (Strategy-tree). *A strategy can be represented as a tree, each node of which represents an atomic attack $\mathcal{X}_j$ and each node may have two or less successors, that correspond to the choice of the next move in two cases $x_j = 0$ and $x_j = 1$. The root node of the strategy-tree represents the first move.*

Definition 6 (Empty strategy). *A strategy S may suggest not to play the attack game of $\mathcal{A}$ at all. Such a strategy can be represented as an empty tree and is denoted by $\emptyset$.*

Definition 7 (Branch of a strategy). *By a* branch *β of the strategy S for an attack tree $\mathcal{A}$, we mean a sequence of assignments*

$$\beta = \langle x_{i_1} = v_1, \ldots, x_{i_{k-1}} = v_{k-1}, x_{i_k} = v_k \rangle \ , \tag{2}$$

where $v_1, \ldots, v_{k-1} \in \{0, 1\}$, $v_k \in \{0, 1, \perp\}$ that may occur when the attack game of $\mathcal{A}$ is played according to S. A branch can also be viewed as a sequence of nodes from the root to a leaf in the strategy-tree together with the outcome v_k of the last-tried atomic attack. Let $\beta \Rightarrow \mathcal{A}$ denote the proposition that the assignments (2) of β imply $\mathcal{F}_{\mathcal{A}}(x_1, \ldots, x_m) = 1$.

Every branch represents a possible sequence of events when playing the attack game with strategy S. We associate to each branch β the following parameters: the cost C_β, the penalty Π_β and the prize P_β. For the branch (2), we have:

$$C_\beta = \sum_{i=1}^{m} C_i \cdot [x_i \in \beta] = C_{i_1} + C_{i_2} + \ldots + C_{i_k}$$

$$\Pi_\beta = \sum_{i=1}^{m} \Pi_i \cdot [x_i \in \beta] \cdot [x_i = \perp] = \begin{cases} \Pi_{i_k} & \text{if } w_k = \perp, \\ 0 & \text{otherwise} \end{cases}$$

$$P_\beta = P \cdot [\beta \Rightarrow \mathcal{A}] = \begin{cases} P \text{ if } x_{i_1} = w_1, \ldots, x_{i_k} = w_k \text{ imply } \mathcal{F}(x_1, \ldots, x_m) = 1, \\ 0 \ \text{ otherwise} \end{cases}$$

where the parentheses [] denote the so-called *Iverson symbol*—for any proposition $\mathcal{P}$, the Iverson symbol $[\mathcal{P}] = 1$ if $\mathcal{P}$ is true, and $[\mathcal{P}] = 0$, otherwise.

Definition 8 (Utility of a strategy). *By the* utility *of a strategy S for an attack tree $\mathcal{A}$, we mean*

$$U(\mathcal{A}; S) = \sum_{\beta} \mathsf{P}[\beta] \cdot (-C_\beta - \Pi_\beta + P_\beta) \ , \tag{3}$$

where $\mathsf{P}[\beta]$ is the probability that β occurs during the attack game played with strategy S. For example, for the branch (2), $\mathsf{P}[\beta] = P_{i_1} \cdot P_{i_2} \cdot \ldots \cdot P_{i_k}$, where

$$P_{i_j} = \begin{cases} p_{i_j} & \text{if } w_j = 1 \\ q_{i_j} & \text{if } w_j = 0 \\ 1 - p_{i_j} - q_{i_j} & \text{if } w_j = \perp \end{cases}$$

For the empty strategy $\emptyset$ we have $U(\mathcal{A}; \emptyset) = 0$ for every attack tree $\emptyset$.

Example: For the attack tree and the strategy of Fig. 1, we have the following seven branches $\beta_1, \ldots, \beta_7$ listed in Tab. 1. Hence, by (3) the utility $U(\mathcal{A}; S)$ of the strategy is computed as follows:

$$\begin{aligned} U(\mathcal{A};S) = {} & p_1p_2 \cdot (-C_1 - C_2 + P) + p_1(1-p_2-q_2) \cdot (-C_1 - C_2 - \Pi_2) + \\ & + p_1q_2p_3 \cdot (-C_1 - C_2 - C_3 + P) + \\ & + p_1q_2(1-p_3-q_3) \cdot (-C_1 - C_2 - C_3 - \Pi_3) - \\ & - p_1q_2q_3 \cdot (C_1 + C_2 + C_3) - (1-p_1-q_1) \cdot (C_1 + \Pi_1) - q_1C_1 \ . \end{aligned}$$

Table 1. The branches of the strategy of Fig. 1 and their parameters

β	Assignments	Probability $\mathsf{P}[\beta]$	Cost C_β	Penalty Π_β	prize P_β
β_1	$x_1 = 1, x_2 = 1$	p_1p_2	$C_1 + C_2$	0	P
β_2	$x_1 = 1, x_2 = \perp$	$p_1(1-p_2-q_2)$	$C_1 + C_2$	Π_2	0
β_3	$x_1 = 1, x_2 = 0, x_3 = 1$	$p_1q_2p_3$	$C_1 + C_2 + C_3$	0	P
β_4	$x_1 = 1, x_2 = 0, x_3 = \perp$	$p_1q_2(1-p_3-q_3)$	$C_1 + C_2 + C_3$	Π_3	0
β_5	$x_1 = 1, x_2 = 0, x_3 = 0$	$p_1q_2q_3$	$C_1 + C_2 + C_3$	0	0
β_6	$x_1 = \perp$	$1 - p_1 - q_1$	C_1	Π_1	0
β_7	$x_1 = 0$	q_1	C_1	0	0

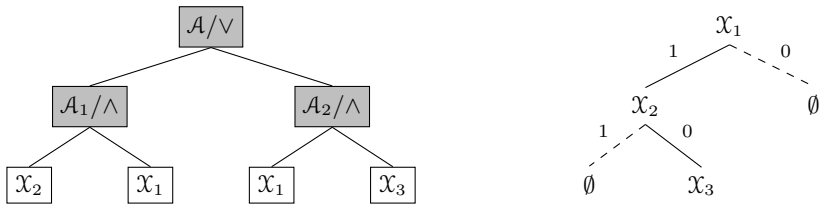

Fig. 1. An attack tree $\mathcal{A}$ (left) and a strategy (right)

Definition 9 (Utility of an attack tree). *By the* utility *of an attack tree $\mathcal{A}$ we mean the limit $U(\mathcal{A}) = \sup_S U(\mathcal{A}; S)$, which exists due to the bound $U(\mathcal{A}; S) \le P$ (where P is the prize) for any strategy S.*

Corollary 1. *$U(\mathcal{A}) \ge 0$ for any $\mathcal{A}$, as $U(\mathcal{A}) = \sup_S U(\mathcal{A}; S) \ge U(\mathcal{A}; \emptyset) = 0$.*

Theorem 1 (Optimal strategy). *For any attack tree $\mathcal{A}$, there exists an* optimal strategy, *i.e. a strategy S for which $U(\mathcal{A}; S) = U(\mathcal{A})$.*

Proof. We use induction on the number m of atomic attacks in $\mathcal{A}$. The statement is clearly true for $m = 0$. Assume that every attack tree $\mathcal{A}'$ with $m - 1$ atomic attacks has an optimal strategy. Let $\mathcal{A}$ be an attack tree with atomic attacks $\mathcal{X}_1, \ldots, \mathcal{X}_m$. Let $S_\mathcal{A}$ be the strategy that first finds $\mathcal{X}_j$ that maximizes the value:

$$u_j = -C_j - (1 - p_j - q_j) \cdot \Pi_j + q_j \cdot U(\mathcal{A}_{x_j=0}) + p_j \cdot U(\mathcal{A}_{x_j=1}) \ ,$$

and chooses $\mathcal{X}_j$ as the next move if $u_j > 0$, or gives up if $u_j \le 0$. After that:

– if $x_j = 0$, $S_{\mathcal{A}}$ uses an optimal strategy S'_0 for $\mathcal{A}_{x_j=0}$ (induction hypothesis);
– if $x_j = 1$, $S_{\mathcal{A}}$ uses an optimal strategy S'_1 for $\mathcal{A}_{x_j=1}$.

Clearly, $S_{\mathcal{A}}$ is optimal for $\mathcal{A}$. □

Corollary 2 (Algorithm for Exact Utility). *The exact utility in the fully adaptive model can be computed by using the following recursive relation*

$$U(\mathcal{A}) = \max_j\{0, -C_j - (1 - p_j - q_j) \cdot \Pi_j + p_j U(\mathcal{A}_{x_j=1}) + q_j U(\mathcal{A}_{x_j=0})\} \ , \quad (4)$$

with initial conditions $U(\mathbf{1}) = P$ and $U(\mathbf{0}) = 0$, where $\mathbf{1}$ and $\mathbf{0}$ denote attack games with Boolean functions $\mathcal{F} \equiv 1$ and $\mathcal{F} \equiv 0$, respectively.

This algorithm runs in time $O(m!)$, where m is the number of atomic attacks, and is hence unsuitable if m is large.

3.3 Simulated Strategies

The concept of simulated strategies is a useful tool for drawing implications about the utility $U(\mathcal{A})$ of an attack tree $\mathcal{A}$ based on the utilities of its subtrees $\mathcal{B}$ (Def. 2). Let $\mathcal{A}$ be an attack tree $\mathcal{A}$ and $\mathcal{B}$ be a subtree of $\mathcal{A}$.

Definition 10 (Simulated strategy). *Every strategy S for $\mathcal{A}$ can be modified to a* simulated strategy $S\,|\,\mathcal{B}$ *for $\mathcal{B}$, in the following way:*

– *Whenever S decides to try an atomic attack in $\mathcal{B}$, then so does strategy $S\,|\,\mathcal{B}$.*
– *If S decides to try an atomic attack $\mathcal{X}_i$ that $\mathcal{B}$ does not involve, then $\mathcal{X}_i$ is* simulated *by $S \mid \mathcal{B}$ (without actually investing into it) and the results are hold as auxiliary information a.*

Let C_β and $C_{\beta|\mathcal{B}}$ be the costs of S and $S\,|\,\mathcal{B}$ respectively in branch β of S. Then

$$C_{\beta|\mathcal{B}} = \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] \cdot [x_j \in \mathcal{B}] \le \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] = C_\beta \ . \quad (5)$$

If Π_β and $\Pi_{\beta|\mathcal{B}}$ are the penalties of S and $S\,|\,\mathcal{B}$ (in case of β) respectively, then

$$\Pi_{\beta|\mathcal{B}} = \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] \cdot [x_j \in \mathcal{B}] \le \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] = \Pi_\beta \ . \quad (6)$$

Let P_β and $P_{\beta|\mathcal{B}}$ be the prize of S and $S \mid \mathcal{B}$ respectively in β. Without any additional assumptions about $\mathcal{A}$ and $\mathcal{B}$, we do not know the relationship between $P_\beta = P \cdot [\beta \Rightarrow \mathcal{A}]$ and $P_{\beta|\mathcal{B}} = P \cdot [\beta \Rightarrow \mathcal{B}]$. Lemma 1 is an important special case.

Lemma 1. *If $\mathcal{B}$ is a subtree of $\mathcal{A}$ and $\mathcal{F}_{\mathcal{A}}(x_1, \ldots, x_m) \Rightarrow \mathcal{F}_{\mathcal{B}}(x_1, \ldots, x_m)$ is a tautology, then $U(\mathcal{A}) \le U(\mathcal{B})$.*

Proof. As $\beta \Rightarrow \mathcal{A}$ implies $\beta \Rightarrow \mathcal{B}$, then $P_\beta \,|\, \mathcal{B} \geq P_\beta$, and by (5) and (6) we have:

$$\begin{aligned} U(\mathcal{A}; S) &= \sum_\beta \mathsf{P}[\beta] \cdot (-C_\beta - \Pi_\beta + P_\beta) \leq \sum_\beta \mathsf{P}[\beta] \cdot (-C_{\beta|\mathcal{B}} - \Pi_{\beta|\mathcal{B}} + P_{\beta|\mathcal{B}}) \\ &= U(\mathcal{B}; S \,|\, \mathcal{B}) \leq U(\mathcal{B}) \ , \end{aligned}$$

for any strategy S for $\mathcal{A}$ and the corresponding simulated strategy for $\mathcal{B}$. Hence, in case S is an optimal strategy for $\mathcal{A}$, we have $U(\mathcal{A}) = U(\mathcal{A}; S) \leq U(\mathcal{B})$. □

4 Efficient Decomposition Rules

4.1 The AND-Rule

An attack tree $\mathcal{A}$ (with Boolean formula $\mathcal{F}$) is a $\wedge$-*composition* of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ (with Boolean formulae $\mathcal{F}_1, \ldots, \mathcal{F}_n$, respectively) and write $\mathcal{A} = \mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n$, if

$$\mathcal{F}(x_1, \ldots, x_m) \equiv \mathcal{F}_1(x_1, \ldots, x_m) \wedge \ldots \wedge \mathcal{F}_n(x_1, \ldots, x_m) \ , \tag{7}$$

where $x_1, \ldots, x_m$ represent the atomic attacks $\mathcal{X}_1, \ldots, \mathcal{X}_m$ that the trees contain.

Theorem 2 (AND-rule). $U(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n) \leq \min\{U(\mathcal{A}_1), U(\mathcal{A}_2), \ldots, U(\mathcal{A}_n)\}$.

Proof. As $\mathcal{F}(x_1, \ldots, x_m) \Rightarrow \mathcal{F}_i(x_1, \ldots, x_m)$ is a tautology for every $i = 1..n$, by Lemma 1, $U(\mathcal{A}) \leq U(\mathcal{A}_i)$ and hence $U(\mathcal{A}) \leq \min\{U(\mathcal{A}_1), \ldots, U(\mathcal{A}_n)\}$. □

Definition 11 (Non-Stop Strategy). *A strategy S for an attack tree $\mathcal{A}$ is called a* non-stop strategy*, if for any branch β, either: (1) $\beta \Rightarrow \mathcal{A}$, (2) $\beta \Rightarrow \neg\mathcal{A}$, or (3) β contains an assignment $(x_j, \bot)$.*

Let $U_{\mathrm{ns}}(\mathcal{A}) = \sup_N U(\mathcal{A}; N)$, where N varies over all non-stop strategies. Clearly, $U_{\mathrm{ns}}(\mathcal{A}) \leq U(\mathcal{A})$.

Lemma 2. *Let $\mathcal{A} = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_m$, where $\mathcal{X}_1, \ldots, \mathcal{X}_m$ are atomic. Then* $U(\mathcal{A}) = \begin{cases} U_{\mathrm{ns}}(\mathcal{A}) & \text{, if } U_{\mathrm{ns}}(\mathcal{A}) > 0 \\ 0 & \text{otherwise.} \end{cases}$

Proof. We use induction on m. The statement is clearly true for $m = 1$. Let $c_i = C_i + (1 - q_i - p_i)\Pi_i$. Assume that the statement is true for $m - 1$ and let $\mathcal{O}$ be an optimal strategy for playing $\mathcal{A} = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_m$. Assume without loss of generality that $U(\mathcal{A}) > 0$ and $\mathcal{X}_m$ was the first atomic attack $\mathcal{O}$ tries. Let $\mathcal{A}' = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_{m-1}$. Hence, $U(\mathcal{A}) = U(\mathcal{A}; \mathcal{O}) = -c_m + p_m U(\mathcal{A}') > 0$, and hence also $U(\mathcal{A}') > 0$. By the induction assumption, $U(\mathcal{A}') = U_{\mathrm{ns}}(\mathcal{A}')$ and hence,

$$U_{\mathrm{ns}}(\mathcal{A}) \leq U(\mathcal{A}) = U(\mathcal{A}; \mathcal{O}) = -c_m + p_m U(\mathcal{A}') = -c_m + p_m U_{\mathrm{ns}}(\mathcal{A}') \leq U_{\mathrm{ns}}(\mathcal{A}) \ ,$$

which implies $U(\mathcal{A}) = U_{\mathrm{ns}}(\mathcal{A})$. □

Theorem 3 (Atomic AND Case). *If $\mathcal{A} = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_m$ then the best first move of the adversary is to try $\mathcal{X}_i$ with the smallest cost-nonsuccess ratio $\frac{c_i}{1-p_i}$.*

Proof. If $U(\mathcal{A}) > 0$ then $U(\mathcal{A}) = U_{\text{ns}}(\mathcal{A})$. Let S and S' be non-stop strategies that are identical except that in S' the k-th and $(k+1)$-th trials are exchanged. Let $c_{1,k-1}$ denote the average cost of the first $k-1$ atomic attacks, which is the same for S and S', let $p_{1,k-1}$ denote the probability that the first $k-1$ trials are all successful, and let $U_{k+2}(\mathcal{A})$ be the utility of $\mathfrak{X}_{k+2} \wedge \mathfrak{X}_{k+3} \wedge \ldots \wedge \mathfrak{X}_m$. As

$$\begin{aligned} U(\mathcal{A}; S) &= -c_{1,k-1} + p_{1,k-1} \cdot [-c_k + p_k(-c_{k+1} + p_{k+1} \cdot U_{k+2}(\mathcal{A}))] \\ &= -c_{1,k-1} + p_{1,k-1} \cdot [-c_k - p_k c_{k+1} + p_k p_{k+1} \cdot U_{k+2}(\mathcal{A})] \\ U(\mathcal{A}; S') &= -c_{1,k-1} + p_{1,k-1} \cdot [-c_{k+1} - p_{k+1} c_k + p_k p_{k+1} \cdot U_{k+2}(\mathcal{A})] \ , \end{aligned}$$

$U(\mathcal{A}; S) > U(\mathcal{A}; S')$ iff $-c_k - p_k c_{k+1} > -c_{k+1} - p_{k+1} c_k$, i.e. $\frac{c_k}{1-p_k} < \frac{c_{k+1}}{1-p_{k+1}}$. □

Corollary 3. *For any attack game of the form $\mathcal{A} = \mathfrak{X}_1 \wedge \ldots \wedge \mathfrak{X}_m$ there is a $O(m \log m)$-time algorithm for finding the optimal order of the atomic attacks, because by Thm. 3, to find the optimal order, we have to sort the atomic attacks by their cost-nonsuccess ratio and sorting requires $O(m \log m)$-time.*

4.2 OR-Rule for Independent Trees

An attack tree $\mathcal{A}$ (with Boolean formula $\mathcal{F}$) is a $\vee$-*composition* of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ (with Boolean formulae $\mathcal{F}_1, \ldots, \mathcal{F}_n$, respectively) and write $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$, if

$$\mathcal{F}(x_1, \ldots, x_m) \equiv \mathcal{F}_1(x_1, \ldots, x_m) \vee \ldots \vee \mathcal{F}_n(x_1, \ldots, x_m) \ , \tag{8}$$

where $x_1, \ldots, x_m$ represent the atomic attacks $\mathfrak{X}_1, \ldots, \mathfrak{X}_m$ that the trees contain.

Definition 12 (Independent Trees). *Attack trees $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are said to be* independent*, if they do not contain common atomic attacks, i.e. if their Boolean formulae $\mathcal{F}_1, \ldots, \mathcal{F}_n$ do not have common variables.*

For example, $\mathcal{F}_1(x_1, \ldots, x_4) = x_1 \wedge x_2$ and $\mathcal{F}_2(x_1, \ldots, x_4) = x_3 \wedge x_4$ are independent but $x_1 \wedge x_2$ and $x_1 \wedge x_3$ (Fig. 1) are not as x_1 is their common variable. It turns out that if $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are independent and negative (i.e. $U(\mathcal{A}_i) \leq 0$ for all i), then also $U(\mathcal{A}) \leq 0$. The independence is necessary for the negativity rule to hold—we give a counter-example with attack trees that are not independent.

Theorem 4 (OR Rule for Independent Trees). *Let $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ where $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are independent. Then $U(\mathcal{A}) \leq U(\mathcal{A}_1) + U(\mathcal{A}_2) + \ldots + U(\mathcal{A}_n)$.*

Proof. Let S be a optimal strategy for $\mathcal{A}$ and $S\,|\,\mathcal{A}_i$ be the simulated strategy for $\mathcal{A}_i$. Due to $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ we have $\sum_{i=1}^n [\beta \Rightarrow \mathcal{A}_i] \geq [\beta \Rightarrow \mathcal{A}]$, and because of independence, $\sum_{i=1}^n [x_j \in \mathcal{A}_i] = 1$ for every variable x_j that $\mathcal{A}$ contains. Hence,

$$\sum_{i=1}^{n} P_{\beta|\mathcal{A}_i} = P \cdot \sum_{i=1}^{n} [\beta \Rightarrow \mathcal{A}_i] \geq P \cdot [\beta \Rightarrow \mathcal{A}] = P_\beta \ ,$$

$$\sum_{i=1}^{n} C_{\beta|\mathcal{A}_i} = \sum_{i=1}^{n} \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] \cdot [x_j \in \mathcal{A}_i] = \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] \cdot \sum_{i=1}^{n} [x_j \in \mathcal{A}_i]$$

$$= \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] = C_\beta \ ,$$

$$\sum_{i=1}^{n} \Pi_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] \cdot \sum_{i=1}^{n} [x_j \in \mathcal{A}_i]$$

$$= \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] = \Pi_\beta \ .$$

Therefore,

$$U(\mathcal{A}) = \sum_{\beta} \mathsf{P}[\beta] \cdot (-C_\beta - \Pi_\beta + P_\beta) \leq \sum_{i=1}^{n} \sum_{\beta} \mathsf{P}[\beta] \cdot (-C_{\beta|\mathcal{A}_i} - \Pi_{\beta|\mathcal{A}_i} + P_{\beta|\mathcal{A}_i})$$

$$= U(\mathcal{A}_1; S\,|\mathcal{A}_1) + \ldots + U(\mathcal{A}_n; S\,|\mathcal{A}_n) \leq U(\mathcal{A}_1) + \ldots + U(\mathcal{A}_n) \ . \qquad \square$$

4.3 Counterexample with Common Atomic Attacks

If in the case $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$, where the attacks may have common atomic attacks. It turns out that the OR-rule (If all $\mathcal{A}_i$ are negative then $\mathcal{A}$ is negative) does not hold in this case. There is the following counter-example. Let $\mathcal{A} = \mathcal{A}_1 \vee \mathcal{A}_2$, where $\mathcal{A}_1 = \mathcal{X}_1 \wedge \mathcal{X}_2$, and $\mathcal{A}_2 = \mathcal{X}_1 \wedge \mathcal{X}_3$ (Fig. 1). Let all the three atomic attacks $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ have the same parameters $c = C + (1-p-q) \cdot \Pi = 1$, $p = q = 0.5$; and let $P = 5$. Then,

$$U(\mathcal{X}_1 \wedge \mathcal{X}_2) = U(\mathcal{X}_1 \wedge \mathcal{X}_3) = -1 + 0.5(-1 + 0.5 \cdot 5) = -1.5 + 1.25 = -0.25 < 0$$

but as $U(\mathcal{X}_1) = U(\mathcal{X}_2) = U(\mathcal{X}_3) = -1 + 0.5 \cdot 5 = 1.5$ and $U(\mathcal{X}_2 \vee \mathcal{X}_3) = U(\mathcal{X}_2) + q \cdot U(\mathcal{X}_3) = (1+q) \cdot 1.5 = 1.5 \cdot 1.5 = 2.25$, we have (by trying $\mathcal{X}_1$ first)

$$U(\mathcal{A}) \geq -c + p \cdot U(\mathcal{X}_2 \vee \mathcal{X}_3) = -1 + 0.5 \cdot 2.25 = 0.125 > 0 \ .$$

This means that in the proof of the OR-rule, we certainly have to assume that the attacks $\mathcal{A}_1, \ldots, \mathcal{A}_n$ do not have common atomic attacks.

4.4 General OR-Rule: Cost Reduction

Instead of proving $U(\mathcal{A}) \leq U(\mathcal{A}_1) + \ldots + U(\mathcal{A}_n)$, which is not true in general, we modify the attack (sub-) trees $\mathcal{A}_1, \ldots \mathcal{A}_n$ by artificially reducing the costs C_j and the penalties Π_j (thereby, making the attacks easier to perform), and prove

$$U(\mathcal{A}) \leq U(\mathcal{A}'_1) + U(\mathcal{A}'_2) + \ldots + U(\mathcal{A}'_n) \ ,$$

where $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ are the modified attack trees. The philosophy behind this is that if the system is secure even if some of the attacks are assumed to be easier then they really are, then also the attacks against the real system are infeasible.

In order to see, how the costs should be reduced, we study the reasons why the inequality $U(\mathcal{A}) \le U(\mathcal{A}_1) + \ldots + U(\mathcal{A}_n)$ fails. If $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are allowed to have common atomic attacks $\mathcal{X}_j$, then for some atomic attacks $\mathcal{X}_j$ we may have $\sum_{i=1}^{n}[x_j \in \mathcal{A}_i] = k_j > 1$, where k_j is the number of attacks among $\mathcal{A}_1, \ldots, \mathcal{A}_n$ that contain $\mathcal{X}_j$. By using the same simulation concept as before, we have:

$$\sum_{i=1}^{n} P_{\beta|\mathcal{A}_i} = P \cdot \sum_{i=1}^{n}[\beta \Rightarrow \mathcal{A}_i] \ge P \cdot [\beta \Rightarrow \mathcal{A}] = P_\beta$$

$$\sum_{i=1}^{n} C_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] \cdot \sum_{i=1}^{n}[x_j \in \mathcal{A}_i] = \sum_{j=1}^{m} k_j C_j \cdot [x_j \in \beta] \not\le C_\beta \quad (9)$$

$$\sum_{i=1}^{n} \Pi_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] \cdot \sum_{i=1}^{n}[x_j \in \mathcal{A}_i] \quad (10)$$

$$= \sum_{j=1}^{m} k_j \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] \not\le \Pi_\beta$$

We see that the main reason for failure is that we have the terms $k_j C_j$ and $k_j \Pi_j$ instead of C_j and Π_j. This inspires the following theorem.

Theorem 5 (Uniform Cost Reduction). *Let $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ and let $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ be sub-trees with the cost and penalties reduced by the rule $C'_j = \frac{C_j}{k_j}$ and $\Pi'_j = \frac{\Pi_j}{k_j}$. Then $U(\mathcal{A}) \le U(\mathcal{A}'_1) + U(\mathcal{A}'_2) + \ldots + U(\mathcal{A}'_n)$.*

Proof. Let S be an optimal strategy for $\mathcal{A}$ and let $S\,|\,\mathcal{A}_i$ be the simulated strategy for $\mathcal{A}'_i$. By (9) we have

$$\sum_{i=1}^{n} C'_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} k_j C'_j \cdot [x_j \in \beta] = \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] = C_\beta$$

$$\sum_{i=1}^{n} \Pi'_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} k_j \Pi'_j \cdot [x_j \in \beta] \cdot [x_j = \bot] = \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] = \Pi_\beta \ .$$

Hence,

$$U(\mathcal{A}) = \sum_{\beta} \mathsf{P}[\beta] \cdot (-C_\beta - \Pi_\beta + P_\beta) \le \sum_{i=1}^{n} \sum_{\beta} \mathsf{P}[\beta] \cdot (-C'_{\beta|\mathcal{A}_i} - \Pi'_{\beta|\mathcal{A}_i} + P'_{\beta|\mathcal{A}_i})$$

$$= U(\mathcal{A}'_1; S\,|\,\mathcal{A}_1) + \ldots + U(\mathcal{A}'_n; S\,|\,\mathcal{A}_n) \le U(\mathcal{A}'_1) + \ldots + U(\mathcal{A}'_n) \ . \quad \square$$

Cost reduction can be generalized so that the costs and penalties in the attack games $\mathcal{A}'_i$ are reduced in different ways, i.e. the reduction amount may depend on i.

Theorem 6 (General Cost Reduction). *Let* $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ *and let* $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ *be sub-trees with the cost and penalties reduced in* $\mathcal{A}'_i$ *by general rules* $C_j \mapsto C'_{j,i}$ *and* $\Pi_j \mapsto \Pi'_{j,i}$*, so that*

$$\sum_{i=1}^{n} C'_{j,i} \cdot [x_j \in \mathcal{A}_i] = C_j \ , \qquad \text{and} \qquad \sum_{i=1}^{n} \Pi'_{j,i} \cdot [x_j \in \mathcal{A}_i] = \Pi_j \ . \tag{11}$$

Then $U(\mathcal{A}) \leq U(\mathcal{A}'_1) + U(\mathcal{A}'_2) + \ldots + U(\mathcal{A}'_n)$*. The Iverson symbol* $[x_j \in \mathcal{A}_i]$ *can be omitted in (11) if we assume that* $C'_{j,i} = \Pi'_{j,i} = 0$ *if* $\mathcal{A}_i$ *does not contain* x_j*.*

Proof. Because of

$$\sum_{i=1}^{n} C'_{\beta|\mathcal{A}_i} = \sum_{i=1}^{n}\sum_{j=1}^{m} C'_{j,i} \cdot [x_j \in \beta] \cdot [x_j \in \mathcal{A}_i] = \sum_{j=1}^{m} [x_j \in \beta] \cdot \underbrace{\sum_{i=1}^{n} C'_{j,i} \cdot [x_j \in \mathcal{A}_i]}_{C_j}$$

$$= C_\beta \ ,$$

$$\sum_{i=1}^{n} \Pi'_{\beta|\mathcal{A}_i} = \sum_{i=1}^{n}\sum_{j=1}^{m} \Pi'_{j,i} \cdot [x_j \in \beta] \cdot [x_j \in \mathcal{A}_i] = \sum_{j=1}^{m} [x_j \in \beta] \cdot \underbrace{\sum_{i=1}^{n} \Pi'_{j,i} \cdot [x_j \in \mathcal{A}_i]}_{\Pi_j}$$

$$= \Pi_\beta \ ,$$

we have $U(\mathcal{A}) \leq U(\mathcal{A}'_1) + U(\mathcal{A}'_2) + \ldots + U(\mathcal{A}'_n)$, like the proof of Thm. 5. □

4.5 Algorithm 1: Iterated AND/OR Rules

The first algorithm uses the AND-rule and the OR-rule at every node of the attack-tree. In order to make the OR-rule work in general case, cost-reduction is used. The cost of every every atomic attack is reduced at every OR-node of the attack tree. The cost reduction step starts from the root vertex and ends in the root vertices. For example, in case we have an attack tree with Boolean formula

$$\mathcal{F} = (\mathcal{X} \wedge \mathcal{X}_1) \vee (\mathcal{X} \wedge \mathcal{X}_2) \vee (((\mathcal{X} \wedge \mathcal{X}_3) \vee (\mathcal{X} \wedge \mathcal{X}_4)) \wedge \mathcal{X}_5) \ ,$$

which is depicted in Fig. 2 (left), the cost c of the atomic attack $\mathcal{X}$ must be used in two places: (1) in the root node we divide the cost by 3; and (2) in the subtree $(\mathcal{X} \wedge \mathcal{X}_3) \vee (\mathcal{X} \wedge \mathcal{X}_4)$, we have to divide the cost again by 2, and hence, the cost c of $\mathcal{X}$ reduces to $c/6$ in this subtree, while in subtrees $\mathcal{X} \wedge \mathcal{X}_1$ and $\mathcal{X} \wedge \mathcal{X}_2$ it reduces to $c/3$.

After the costs of all atomic attacks are reduced in this way, we apply the AND, OR negativity rules starting from the leaves of the tree (the atomic attacks) and ending with the root vertex.

The algorithm works in time which is roughly linear in the number of vertices of the tree, and hence is very fast. The main drawback of this algorithm is that in case of "deep" attack trees where atomic attacks appear many times, the cost reduction rules will reduce the cost too much, i.e. it is practically impossible to find practical security measures and apply them in the real system so that the system is still secure if reduced costs are assumed.

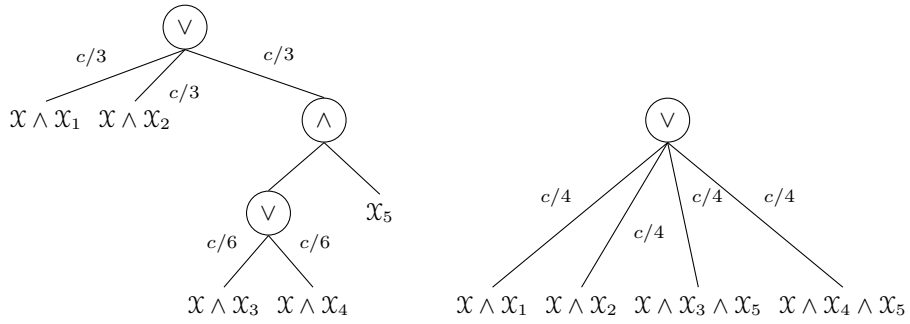

Fig. 2. Iterated cost reduction based on the tree structure (left) and cost reduction based on the DNF (right)

4.6 Algorithm 2: DNF with Cost Reduction

The second algorithm first constructs a DNF of the Boolean function of the attack tree, reduces the cost of each atomic attack by dividing it by the number of min-terms in which the atomic attack appears. For example, if an atomic attack appears in four min-terms of the DNF, then its cost c is reduced to $c/4$ (Fig. 2, right). Then, for each min-term, Algorithm 2 finds the optimal order of the attack by arranging the atomic attacks by their cost-nonsuccess ratio (Theorem 3). Finally, it applies the OR-rule, i.e. checks if all min-terms are of negative utility.

The second algorithm has running time $O(2^m)$, were m is the number of atomic attacks, because constructing a DNF is of exponential complexity. However, it terms of practical applicability, the second algorithm has a big advantage over the first one. The reason is how the min-terms are handled by the two algorithms. To imply the negative utility of a min-term, the first algorithm has to find an atomic attack in the min-term that is of negative utility (and then apply the AND-rule). Such an atomic attack may not exist in the min-term, while the min-term as a whole still has negative utility. The second algorithm computes the exact utility of the min-term instead and hence determines the negativity of min-terms without errors.

The main drawback of Algorithm 2 is that an atomic attack may appear in a considerable fraction of the (exponentially large) set of min-terms and hence, its cost may reduce from a high value to a negligible one.

5 Infinite Repetition Model

The models proposed so far assume that the atomic attacks are tried just once by the adversary. This may not be the case in the real world. If an adversary fails with an atomic attack $\mathcal{X}_j$, then it might be that $\mathcal{X}_j$ can be tried again by the adversary, i.e. in case of failure (with probability q_j) no assignments are made and the position of the attack game remains the same. Such a game is called

attack game with repetition. Note that repetition only makes the attacks easier, i.e. if $U_\infty(\mathcal{A})$ denotes the utility of $\mathcal{A}$ in an attack game with repetition, then $U(\mathcal{A}) \leq U_\infty(\mathcal{A})$. Hence, if we manage to prove that $U_\infty(\mathcal{A}) \leq 0$, then it implies $U(\mathcal{A}) \leq 0$. Hence, it is safe to assume that repetition is always allowed.

5.1 Conversion to Infinite Repetition and Failure-Free Models

If an optimal strategy S chooses an atomic attack $\mathcal{X}$ as the next move and fails then the game remains the same and S chooses $\mathcal{X}$ again. Hence, any atomic attack is iterated until success or getting caught.

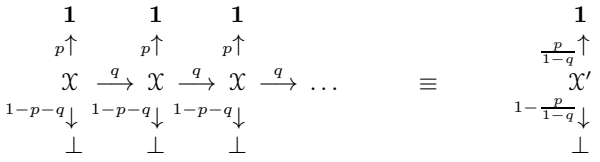

Fig. 3. A repeatable atomic attack $\mathcal{X}$ with parameters (p, q, C, Π) is equivalent to a non-repeatable attack $\mathcal{X}'$ with parameters $(\frac{p}{1-q}, 0, \frac{C}{1-q}, \Pi)$

Theorem 7. *Every repeatable atomic attack $\mathcal{X}$ with parameters (p, q, C, Π) can be replaced with a unrepeatable atomic attack $\mathcal{X}'$ with parameters $(\frac{p}{1-q}, 0, \frac{C}{1-q}, \Pi)$ without changing the utility of the game.*

Proof. The winning probability p' while "iterating" $\mathcal{X}$ (Fig. 3) is $p' = p + qp + q^2p + \ldots = p \cdot (1 + q + q^2 + \ldots) = \frac{p}{1-q}$, because the success probability of the first trial is p, the probability of success at the second trial is $q \cdot p$, the prob. that we win at the third trial is $q^2 \cdot p$, etc. The average cost C' during the iteration is

$$C' = (1-q) \cdot C + q(1-q) \cdot 2C + q^2(1-q) \cdot 3C + \ldots = C + qC + q^2C + \ldots = \frac{C}{1-q} \ .$$

Indeed, the probability that the game ends (with win or penalty) at the first trial is $1-q$, the probability of stopping at the second trial is $q(1-q)$, etc. If the game ends at the first trial, the costs are C. If the game ends at the second trial, the costs are $2C$, etc. The event that the game never ends is a null event. □

Hence, if we apply such a transformation to all atomic attacks, we get a so-called *failure-free* attack tree, in which $q_j = 0$ every atomic attack $\mathcal{X}_j$.

Theorem 8. *Failure-free attack trees $\mathcal{A}$ have optimal strategies that are non-adaptive, i.e. there is a fixed ordering $\mathcal{X}_{i_1}, \ldots, \mathcal{X}_{i_k}$ of atomic attacks that the optimal strategy follows. Moreover, the formula $x_{i_1} \wedge \ldots \wedge x_{i_k}$ is a critical min-term of the Boolean formula $\mathcal{F}$ of $\mathcal{A}$.*

Proof. Let S be an optimal strategy for $\mathcal{A}$ and $\mathcal{X}_{i_1}$ be the best move. As $\mathcal{A}$ is failure-free, there may be two possible outcomes of trying $\mathcal{X}_{i_1}$:

– the adversary "gets caught", the game ends;
– $\mathcal{X}_{i_1}$ is successful and the next game to play is $\mathcal{A}_{x_{i_1}=1}$.

Hence, if the game does not end in the first move, the next move to play is the best move x_{i_2} of the game $\mathcal{A}_{x_{i_1}=1}$. Let $(\mathcal{X}_{i_1},\ldots,\mathcal{X}_{i_k})$ be the order of trials suggested by S. The formula $x_{i_1}\wedge\ldots\wedge x_{i_k}$ is a min-term of $\mathcal{F}$ (the Boolean formula of $\mathcal{A}$), because otherwise S never wins and $U(\mathcal{A})$ would be negative, which is impossible due to Corollary 1 of Thm. 1. If there is x_{x_j} such that $x_{i_1}\wedge\ldots\wedge x_{i_{j-1}}\wedge x_{i_{j+1}}\wedge\ldots\wedge x_{i_k}$ is still a min-term of $\mathcal{F}$, the atomic attack $\mathcal{X}_{i_j}$ can be skipped and we have a strategy S' with $U(S')>U(S)$, which is impossible because S is optimal. Hence, $x_{i_1}\wedge\ldots\wedge x_{i_k}$ is a critical min-term. □

5.2 Algorithm 3: Exact Utility in the Infinite Repetition Model

Theorem 9. *The exact utility in the fully adaptive model can be computed as follows: (1) find the atomic attack $\mathcal{X}$ with the smallest ratio $\frac{c}{1-q-p}$ (where $c=C+(1-q-p)\Pi$); and (2) compute recursively*

$$U_\infty(\mathcal{A})=\max\left\{0,\ \ \frac{-c}{1-q}+\frac{p}{1-q}\cdot U_\infty(\mathcal{A}_{x=1}),\ \ U_\infty(\mathcal{A}_{x=0})\right\}\ \ , \qquad (12)$$

with initial conditions $U_\infty(\mathbf{1})=P$ and $U_\infty(\mathbf{0})=0$, where $\mathbf{1}$ and $\mathbf{0}$ denote attack games with Boolean functions $\mathcal{F}\equiv 1$ and $\mathcal{F}\equiv 0$, respectively.

Proof. If $U_\infty(\mathcal{A})>0$, then by Theorems 3, 7, 8, every optimal strategy in the infinite repetition model is non-adaptive and associated with a critical min-term $x_{i_1}\wedge\ldots\wedge x_{i_k}$ of the corresponding Boolean function and the first move of which is to try the atomic attack $\mathcal{X}\in\{\mathcal{X}_{i_1},\ldots,\mathcal{X}_{i_k}\}$ with the smallest cost-nonsuccess ratio $\frac{c'}{1-p'}=\frac{c}{1-q-p}$, where $p'=\frac{p}{1-q}$ and $c'=\frac{c}{1-q}$ are the transformed parameters. So, it is sufficient to generate all critical min-terms the variables of which are ordered according to the ratios of the corresponding atomic games. If the optimal strategy is a min-term that contains the atomic game $\mathcal{X}$ with the smallest ratio, then $\mathcal{X}$ is the first move and the utility is $\frac{-c}{1-q}+\frac{p}{1-q}\cdot U_\infty(\mathcal{A}_{x=1})$. If the optimal strategy does not involve $\mathcal{X}$, then the utility is $U_\infty(\mathcal{A}_{x=0})$. □

Algorithm 3 runs in time $O(2^m)$, where m is the number of atomic games, and is much more efficient compared to the algorithm that finds exact outcome in the fully adaptive model without repetition. While having approximately the same complexity as Algorithm 2, Algorithm 3 seems to have a big advantage, because it avoids the most important drawback of Algorithm 2—the change of parameters in Algorithm 3 is moderate and does not depend on the DNF size.

6 Open Questions and Further Work

Algorithm 3 has several advantages over the previous methods of estimating adversaries' utility: it has the same complexity than the parallel algorithm of

Jürgenson-Willemson [8] but the bound it gives is much more reliable. Still, the exponential complexity is unsuitable in many practical cases where attack-trees are large. It would be interesting to study algorithms that combine the AND-OR rules (for larger trees) and Algorithm 3 for sufficiently small subtrees of the attack-tree. Such an approach seems promising because attack trees for real systems are "modular", i.e. they consist of subtrees of moderate size that are relatively independent of each other (contain a small number of common atomic attacks). It might be the case that there are better AND-OR rules in the infinite repetition model than in the model without repetition. This needs more research.

Even having some outstanding qualities the model we propose still relies on the ability of analysts to construct precise attack trees that capture all attack vectors. If some atomic attacks are forgotten and not included in the attack tree, they may define a profitable attack suite, while the answer given by the model may imply that the system is secure. This means that security has to be a cyclic process when the list of threats and vulnerabilities is revised constantly.

All attack tree models depend on the metrics assigned to the leaves of the attack tree. Unfortunately there are no good frameworks for metric estimation. Even though some effort has been made to establish methods [4,12,14] for metric calculation, a lot of work has to be done in this field before quantitative attack tree models become as useful as they potentially can be.

References

1. Buldas, A., Laud, P., Priisalu, J., Saarepera, M., Willemson, J.: Rational Choice of Security Measures Via Multi-parameter Attack Trees. In: López, J. (ed.) CRITIS 2006. LNCS, vol. 4347, pp. 235–248. Springer, Heidelberg (2006)
2. Buldas, A., Mägi, T.: Practical Security Analysis of E-Voting Systems. In: Miyaji, A., Kikuchi, H., Rannenberg, K. (eds.) IWSEC 2007. LNCS, vol. 4752, pp. 320–335. Springer, Heidelberg (2007)
3. Convery, S., Cook, D., Franz, M.: An attack tree for the Border Gateway Protocol (2004)
4. Downs, D.D., Haddad, R.: Penetration testing—the gold standard for security rating and ranking. In: Proceedings of the 1st Workshop on Information-Security-System Rating and Ranking (WISSRR), Williamsburg, Virginia, USA (2001)
5. Edge, K.S.: A framework for analyzing and mitigating the vulnerabilities of complex systems via attack and protection trees. Ph.D. thesis, Air Force Institute of Technology, Ohio (2007)
6. Ericson, C.: Fault tree analysis—a history. In: The 17th International System Safety Conference (1999)
7. Jürgenson, A., Willemson, J.: Serial Model for Attack Tree Computations. In: Lee, D., Hong, S. (eds.) ICISC 2009. LNCS, vol. 5984, pp. 118–128. Springer, Heidelberg (2010)
8. Jürgenson, A., Willemson, J.: Computing Exact Outcomes of Multi-parameter Attack Trees. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 1036–1051. Springer, Heidelberg (2008)
9. Mauw, S., Oostdijk, M.: Foundations of Attack Trees. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 186–198. Springer, Heidelberg (2006)

10. Niitsoo, M.: Optimal Adversary Behavior for the Serial Model of Financial Attack Trees. In: Echizen, I., Kunihiro, N., Sasaki, R. (eds.) IWSEC 2010. LNCS, vol. 6434, pp. 354–370. Springer, Heidelberg (2010)
11. Schneier, B.: Attack trees: Modeling security threats. Dr. Dobbs Journal 24(12), 21–29 (1999)
12. Schudel, G., Wood, B.: Adversary Work Factor As a Metric for Information Assurance. In: Proceedings of the 2000 Workshop on New Security Paradigms, Ballycotton, County Cork, Ireland, pp. 23–30 (2000)
13. Weiss, J.D.: A system security engineering process. In: Proc. of the 14th National Computer Security Conf., pp. 572–581 (1991)
14. Wood, B., Bouchard, J.: Read team work factor as a security measurement. In: Proc. of the 1st Workshop on Information-Security-System Rating and Ranking (WISSRR 2001), Williamsburg, Virginia, USA (2001)

A Computational Examples

To show how the proposed models may be applied, we analyze the attack tree of Fig. 1 with the parameters given in Tab. 2. We assume that $\mathcal{X}_1$ can be repeated multiple times, and let the prize be $P = 80$.

Table 2. Atomic attack parameters of the attack tree shown in Fig. 1

Atomic attack	Cost	Success probability	Failure probability	Repeatability
$\mathcal{X}_1$	$c_1 = 6$	$p_1 = 0.2$	$q_1 = 0.2$	Repeatable
$\mathcal{X}_2$	$c_2 = 6$	$p_2 = 0.2$	$q_2 = 0.2$	Non-repeatable
$\mathcal{X}_3$	$c_3 = 3$	$p_3 = 0.2$	$q_3 = 0.2$	Non-repeatable

A.1 Uniform Cost Reduction

First, the repeatable atomic attack $\mathcal{X}_1$ has to be substituted with unrepeatable version $\mathcal{X}_1'$ of itself with the parameters $c_1' = \frac{c_1}{1-q_1} = \frac{6}{0.8} = 7.5$, $p_1' = \frac{p_1}{1-q_1} = \frac{0.2}{0.8} = 0.25$. As $\mathcal{X}_1$ is involved in both $\mathcal{A}_1$ and $\mathcal{A}_2$, the cost of $\mathcal{X}_1'$ is reduced according to the rules of uniform cost reduction $c_1'^{red} = \frac{c_1'}{2} = 3.75$, producing the atomic attack $\mathcal{X}_1'^{red}$. The atomic attacks in $\mathcal{A}_1' = \mathcal{X}_1'^{red} \wedge \mathcal{X}_2$ and $\mathcal{A}_2' = \mathcal{X}_1'^{red} \wedge \mathcal{X}_3$ have to be sorted according to the increasing cost-nonsuccess ratio.

$$\frac{c_1'^{red}}{1-p_1'} = \frac{3.75}{0.75} = 5 \ , \quad \frac{c_2}{1-p_2} = \frac{6}{0.8} = 7.5 \ , \quad \frac{c_3}{1-p_3} = \frac{3}{0.8} = 3.75 \ .$$

In $\mathcal{X}_1'$, the attack $\mathcal{X}_1'^{red}$ must be tried first (because $5 < 7.5$), and in $\mathcal{X}_2'$, we have to try $\mathcal{X}_3$ first (because $3.75 < 5$). Hence,

$$U(\mathcal{A}_1) = -c_1'^{red} + p_1'(-c_2 + p_2 P) = -3.75 + 0.25(-6 + 0.2 \cdot 80) = -1.25 \ ,$$
$$U(\mathcal{A}_2) = -c_3 + p_3(-c_1'^{red} + p_1' P) = -3 + 0.2(-3.75 + 0.25 \cdot 80) = 0.25 \ .$$

Since $U(\mathcal{A}_2) > 0$, according to the OR-rule of attack trees, $\mathcal{A}$ may have positive utility and a profitable attack suite.

A.2 Non-Uniform Cost Reduction

We reduce c'_1 in $\mathcal{A}_1$ to $\frac{c'_1}{3}$ and in $\mathcal{A}_2$ to $\frac{2c'_1}{3}$. The ratio $\frac{c'^{red}_1}{1-p'_1}$ will then be $\frac{10}{3} \approx 3.33$ in $\mathcal{A}'_1$ and $\frac{20}{3} \approx 6.67$ in $\mathcal{A}'_2$. This means that the optimal order of atomic attacks in $\mathcal{A}'_1$ and $\mathcal{A}'_2$ remains the same as in the uniform cost reduction, and we have:

$$U(\mathcal{A}_1) = -c'^{red}_1 + p'_1(-c_2 + p_2 P) = -7.5/3 + 0.25(-6 + 0.2 \cdot 80) = 0 \ ,$$
$$U(\mathcal{A}_2) = -c_3 + p_3(-c'^{red}_1 + p'_1 P) = -3 + 0.2(-5 + 0.25 \cdot 80) = 0 \ .$$

The non-uniform cost reduction shows that there are no utilities larger than zero and no beneficial attacking strategies, so the system which is being analyzed is secure against rational adversaries. Since by using the reduced costs we give additional power to adversaries, using non-uniform cost reduction means that we control how this additional power is redistributed, however we still get an artificial state which is more favorable to the adversary than the original one, hence it gives a valid result.

A.3 Infinite Repetition Model

All atomic attacks $\mathfrak{X}_1, \mathfrak{X}_2, \mathfrak{X}_3$ must be substituted with multiple repetition versions of themselves $\mathfrak{X}'_1, \mathfrak{X}'_2, \mathfrak{X}'_3$ with the following parameters:

$$c'_1 = c'_2 = \frac{6}{0.8} = 7.5, \quad c'_3 = \frac{3}{0.8} = 3.75, \quad p'_1 = p'_2 = p'_3 = \frac{0.2}{0.8} = 0.25 \ .$$

The ratios of the atomic attacks are the following:

$$\frac{c'_1}{1-p'_1} = \frac{7.5}{0.75} = 10 \ , \ \frac{c'_2}{1-p'_2} = \frac{7.5}{0.75} = 10 \ , \ \frac{c'_3}{1-p'_3} = \frac{3.75}{0.75} = 5 \ .$$

Hence, the utility of $\mathcal{A}_1$ and $\mathcal{A}_2$ is computed as follows:

$$U_\infty(\mathcal{A}_1) = -7.5 + 0.25(-7.5 + 0.25 \cdot 80) = -4.375 \ ,$$
$$U_\infty(\mathcal{A}_2) = -3.75 + 0.25(-7.5 + 0.25 \cdot 80) = -0.625 \ .$$

Since $U_\infty(\mathcal{A}_2) = -0.625$ is the larger of the two utilities, this means that $U_\infty(\mathcal{A}) = -0.625$ and it represents an upper bound of the utility $U(\mathcal{A})$ of the original attack tree $\mathcal{A}$ (where $\mathfrak{X}_2$ and $\mathfrak{X}_3$ are not repeatable. Hence, $U(\mathcal{A})$ must also be negative.