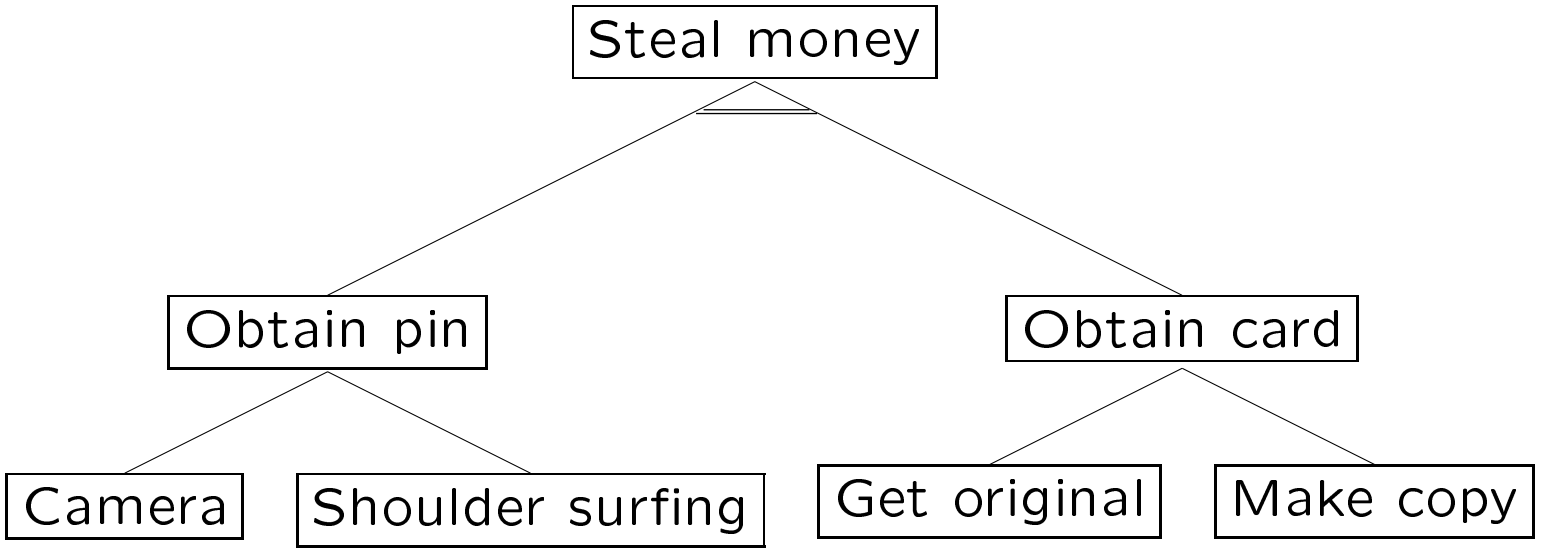
Practical security analysis: The distribution of attack trees

Overview

- What is an attack tree?

- AND-nodes, OR-nodes



- Ahto Buldas, Peeter Laud, Jaan Priisalu, Märt Saarepera, Jan Willemson article *"Rational Choice of Security Measures via Multi-Parameter Attack Trees"*

- Parameters:
  - Gains - the gains of the attacker, in case the attack succeeds
  - Costs - the cost of the attack
  - p - the success probability of the attack
  - π - the average penalty of an attacker if the attack was successuful
  - $\pi_{\_}$ - the average penalty of an attacker if the attack was not successuful

- $\text{Outcome} = -\text{Costs} + p \cdot (\text{Gains} - \pi) - \overline{p} \cdot \pi_{-}$, where $\overline{p} = 1 - p$.

- If we have an OR-node, then:

$$(\text{Costs}, p, \pi, \pi_{-}) = \begin{cases} (\text{Costs}_1, p_1, \pi_1, \pi_{1-}), \text{if } \text{Outcome}_1 > \text{Outcome}_2 \\ (\text{Costs}_2, p_2, \pi_2, \pi_{2-}), \text{if } \text{Outcome}_1 \leq \text{Outcome}_2 \end{cases}$$

where $\text{Outcome}_i = -\text{Costs}_i + p_i \cdot (\text{Gains}_i - \pi_i) - \overline{p}_i \cdot \pi_{i-}, \ \ i = 1, 2.$

- If we have an AND-node, then:

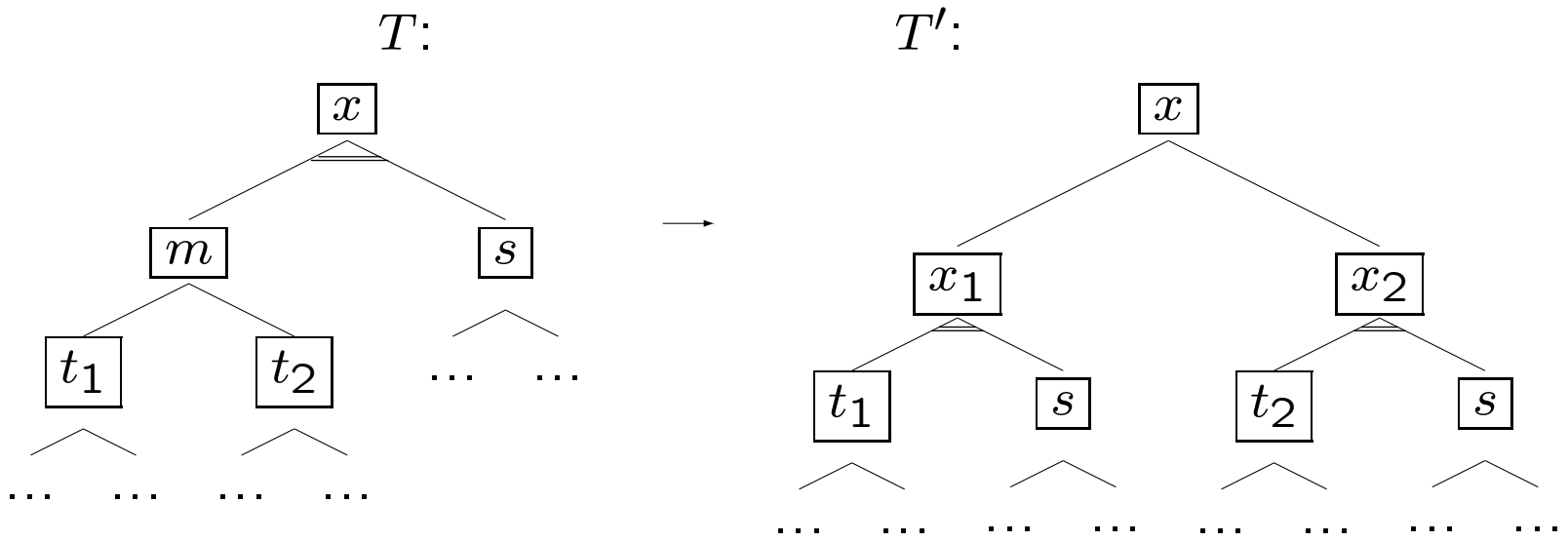
$$\text{Costs} = \text{Costs}_1 + \text{Costs}_2, \quad p = p_1 \cdot p_2, \quad \pi = \pi_1 + \pi_2$$

$$\pi_{-} = \frac{p_1 \overline{p_2} (\pi_1 + \pi_{2-}) + \overline{p_1} p_2 (\pi_{1-} + \pi_2) + \overline{p_1 p_2} (\pi_{1-} + \pi_{2-})}{1 - p_1 p_2}$$

- Sjouke Mauw and Martijn Oostdijk reduction rules (DNF)

$$(t_1 \vee t_2) \wedge s \longmapsto (t_1 \wedge s) \vee (t_2 \wedge s)$$

- Example:



- Sometimes Outcome($T.x$) $\neq$ Outcome($T'.x$)

Let $t_y \in \{t_1, t_2\}$ and

$$C_z = \text{Costs}_z + p_z \cdot \pi_z + (1 - p_z)\pi_{z-}$$

$$\Delta\text{Outcome}_{t_y}(x) = \text{Outcome}_x - \text{Outcome}_s =$$

$$= -C_{t_y} - C_s + p_{t_y} \cdot p_s \cdot \text{Gains} - (-C_s + p_s \cdot \text{Gains}) = -C_{t_y} - (1 - p_{t_y}) \cdot p_s \cdot \text{Gains}$$

$\Delta\text{Outcome}_{t_y}$ as a linear function of p_s, that could be noted as a pair (C_{t_y}, p_{t_y}).

Definitions:

(C_{t_1}, p_{t_1}) *dominates* (C_{t_2}, p_{t_2}) if $C_{t_1} \leq C_{t_2}$ and $p_{t_1} \geq p_{t_2}$.

(C_{t_1}, p_{t_1}) and (C_{t_2}, p_{t_2}) *multidominates* (C_{t_3}, p_{t_3}) if $C_{t_1} < C_{t_3} < C_{t_2}$, $p_{t_1} < p_{t_3} < p_{t_2}$ and the crossing-point of functions of t_1 and t_2 is always higher than the function of t_3

The list method:

Every leaf t has a list of one element (C_t, p_t)

OR-node list contains all its children's lists, where has been eliminated the pairs that are dominated or multidominated

In the AND-node we find every possibility to form the fullattack and then find the parameters in the original way:

$$C = C_1 + C_2 + \ldots + C_n$$

$$p = p_1 \cdot p_2 \cdot \ldots \cdot p_n$$

If (C_{t_1}, p_{t_1}) dominates (C_{t_2}, p_{t_2}), then in AND-node $C_{t_1} + C_{t_3} \leq C_{t_2} + C_{t_3}$ and $p_{t_1} \cdot p_{t_3} \geq p_{t_2} \cdot p_{t_3}$, because $p_{t_3} \geq 0$. If (C_{t_1}, p_{t_1}) and (C_{t_2}, p_{t_2}) multidominates (C_{t_3}, p_{t_3}), then in AND-node $C_{t_1} + C_{t_4} \leq C_{t_3} + C_{t_4} \leq C_{t_2} + C_{t_4}$ and $p_{t_1} \cdot p_{t_4} \leq p_{t_3} \cdot p_{t_4} \leq p_{t_2} \cdot p_{t_4}$, because $p_{t_4} \geq 0$.

Example

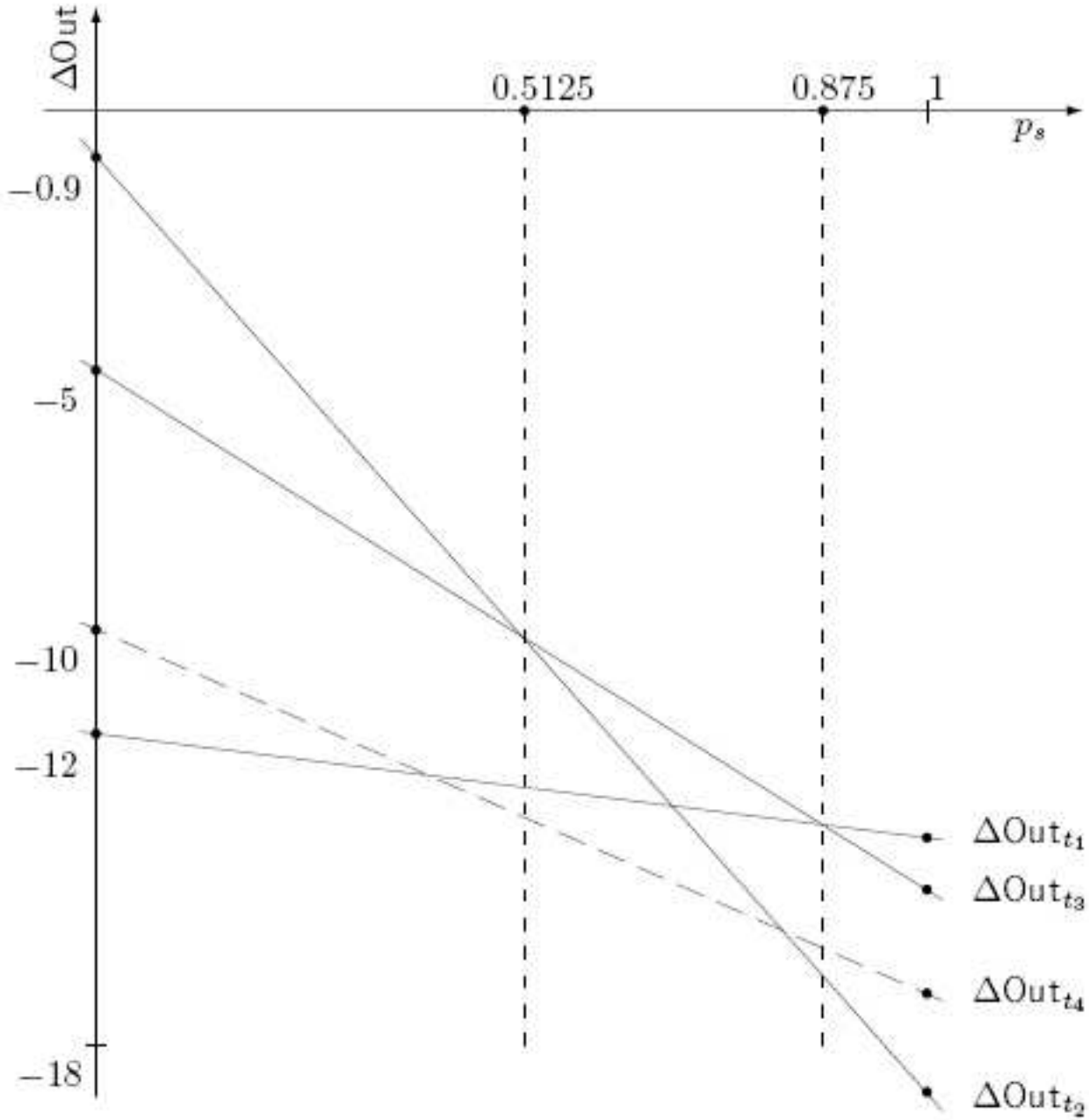
Let $t_1 = (\$0; 0,9; \$0; \$120)$, $t_2 = (\$0; 0,1; \$0; \$1)$, $t_3 = (\$0; 0,5; \$0; \$10)$
$t_4 = (\$10; 0,7; \$0; \$0)$ and $s = (\$0; 0,5; \$0; \$0)$, Gains = \$20.
Then

$$\Delta \text{Out}_{t_1} = -12 - 2p_s$$

$$\Delta \text{Out}_{t_2} = -0.9 - 18p_s$$

$$\Delta \text{Out}_{t_3} = -5 - 10p_s$$

$$\Delta \text{Out}_{t_4} = -10 - 7p_s.$$


ΔOut
0.5125
0.875
1
p_s
−0.9
−5
−10
−12
−18
ΔOut_{t_1}
ΔOut_{t_3}
ΔOut_{t_4}
ΔOut_{t_2}

Estimation method:

- We overestimate the attacker
- Less accurate
- error M and range $[A, B] \in [0, 1]$
- AND-node's parameters C and p is calculated like before, only $M = M_1 + M_2 + \ldots + M_n$
- In OR-node we find (C_1, p_1, M_1) and (C_2, p_2, M_2), where $\Delta\text{Out}_1(A)$ and $\Delta\text{Out}_2(B)$ are the highest on values A and B accordingly

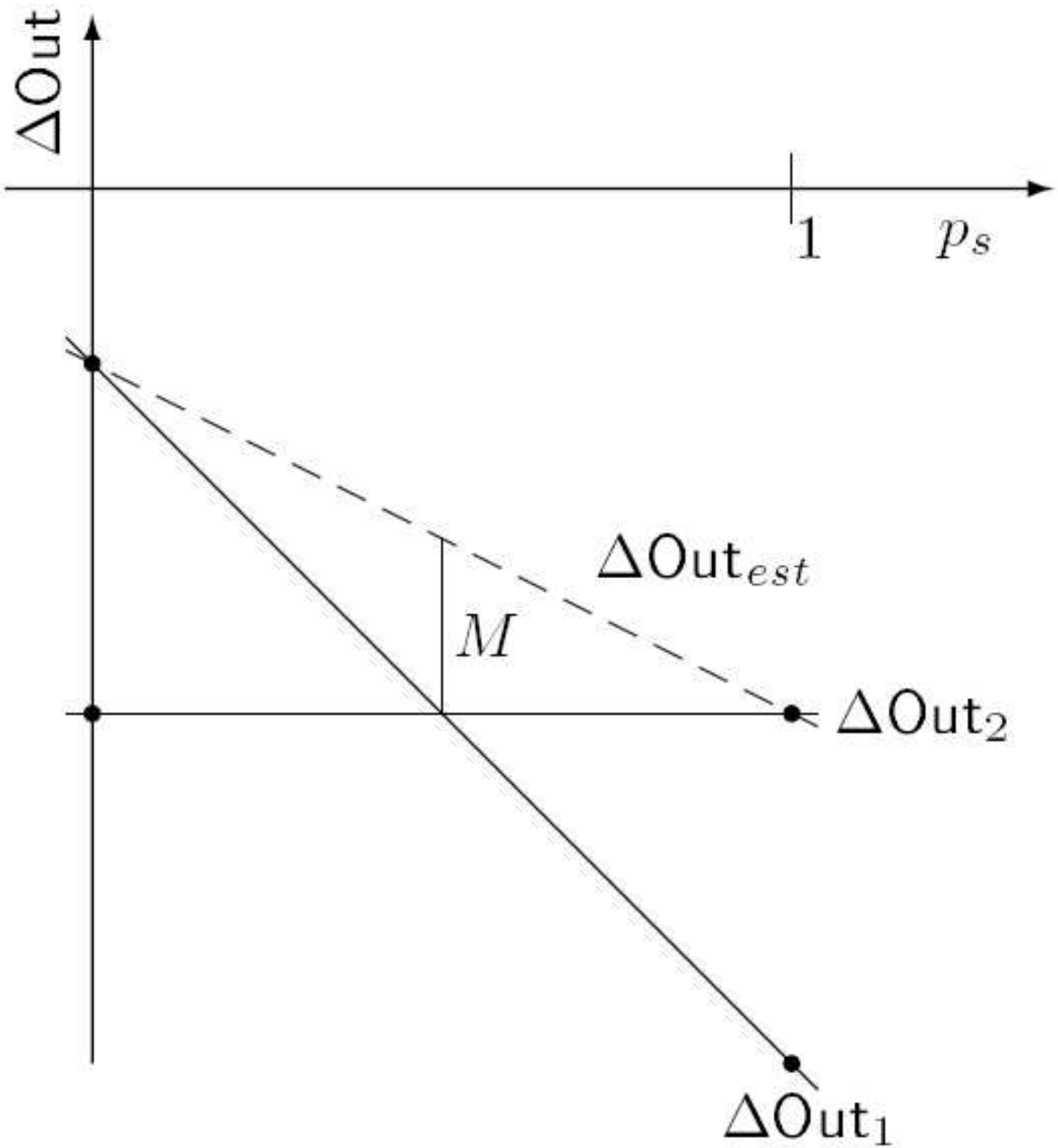
- Generate new function $\Delta\text{Out}_{est} = -C - (1-p) \cdot \text{Gains} \cdot p_s$, where C and p are calculated from $\Delta\text{Out}_1(A)$ and $\Delta\text{Out}_2(B)$

$$p = 1 - \frac{\Delta\text{Out}_2(B) - \Delta\text{Out}_2(A)}{(B-A) \cdot \text{Gains}}$$

$$C = \Delta\text{Out}_1(A) - (1-p) \cdot \text{Gains}$$

- Calculate $M = max(M_1, M_2) + \Delta\text{Out}_{est}(P) - \Delta\text{Out}_1(P)$, P is the crossing-point of $\Delta\text{Out}_1(A)$ and $\Delta\text{Out}_2(B)$

$$P = \frac{C_2 - C_1}{\text{Gains} \cdot (p_2 - p_1)}$$


ΔOut
1
p_s
ΔOut$_{est}$
M
ΔOut$_2$
ΔOut$_1$