Cascade *Cryptanalysis Exemple*

Eric Brier[1], Eric Freysinnet[2], David Naccache[3], Maxime Legrand[3], Thomas Magnard[3], Marc Pasqualetto[3], and Nissim Zerbib[3]

[1] Ingenico
1, rue Claude Chappe, BP 346, F-07503 Guilherand-Granges, France
eric.brier@ingenico.com
[2] LIP6, CNRS et Universit Pierre et Marie Curie
4 place Jussieu - BC 169 F-75252 Paris CEDEX 05, France
eric.freyssinet@botnets.fr
[3] École normale supérieure
Équipe de cryptographie, 45 rue d'Ulm, F-75230 Paris CEDEX 05, France
david.naccache@ens.fr, legrand.maxime@ens.fr, thomas.magnard@ens.fr, marc.pasqualetto@ens.fr, nissim.zerbib@ens.fr

Abstract. ...

1 Introduction

Botnets are networks of infected end-hosts called bots that are under the control of a human operator commonly known as the botmaster (hereafter Alice). While botnets recruit vulnerable machines using methods also utilized by other classes of malware (*e.g.*, remotely exploiting software vulnerabilities, social engineering, etc.), their defining characteristic is the use of command and control (C&C) channels to connect bots to their botmasters. Bot owners are usually unaware that their computers were hijacked to forward transmissions (*e.g.* spam or viruses) to other potential victims on the Internet.

Computers hijacked into a botnet are often those whose owners fail to adequately protect. A bot is often created through an Internet port that has been left open and through which a malicious program can sneak in for future activation. At a certain time, Alice can unleash the effects of the botnet by sending a single command, possibly from an Internet Relay Channel (IRC) site.

We model botnets in a very simple way: each target computer $\mathcal{V}_i$ is assimilated to a black-box having three attributes denoted ϵ_i, w_i, π_i. $\mathcal{V}_i$ can be successfully attacked with probability ϵ_i by executing w_i instructions. As in the Japanese go game (or the *World War Z* movie), as soon as $\mathcal{V}_i$ is conquered, $\mathcal{V}_i$ can be used to attack other targets. The computational power provided by a conquered $\mathcal{V}_i$ is π_i instructions per second. Alice's initial computational power is A[4].

This article attempts to clarify the following question:

> *Given n potential targets $\mathcal{V}_1, \ldots, \mathcal{V}_n$, how should Alice schedule her attacks to conquer all possible targets as quickly as possible?*

[4] $A = 1$ if the unit in which the π_i are expressed is not instructions per second but "Alice-equivalents".

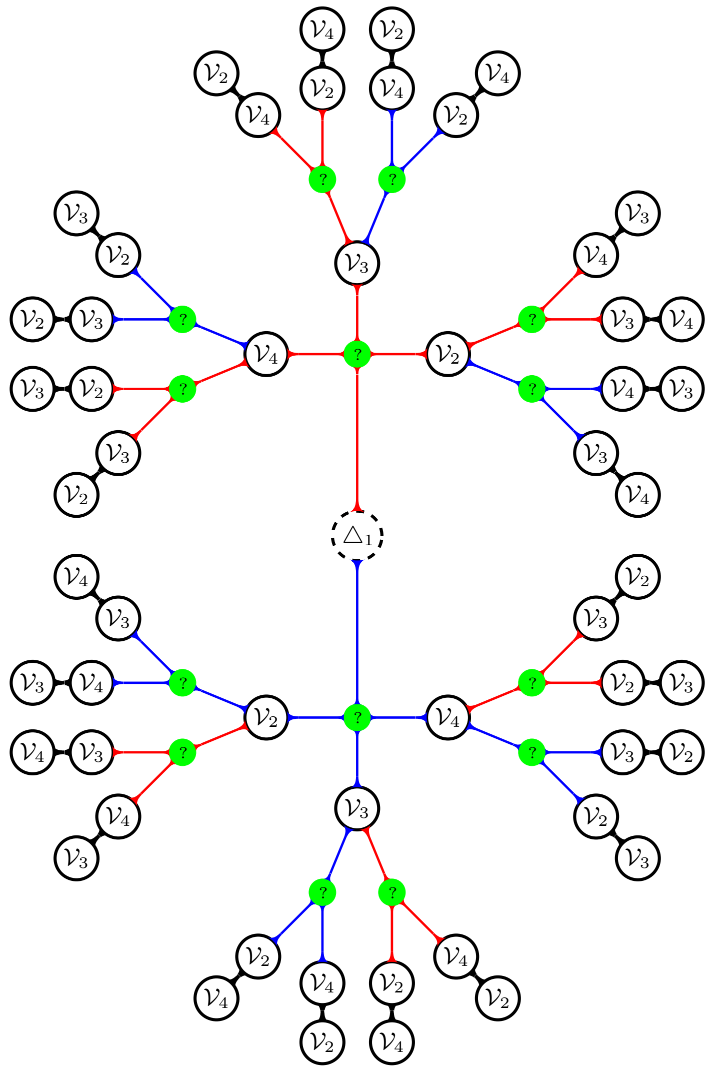


Fig. 1. Finding an adaptive P_{opt} for $n = 4$ (Assuming that it is known that the attack should start by $\mathcal{V}_1$).

Every attack plan P has an expected completion time $\Delta(P)$.

P can be *deterministic* or *adaptive*. In a deterministic attack plan, Alice must define in advance the instant at which each $\mathcal{V}_i$ will be attacked. Attacks will be automatically launched as planned regardless successes or failures. An adaptive attacker has more freedom and progressively adapts her actions to observed successes or failures.

In a deterministic *sequential* plan, P is a permutation of $\{1, \ldots, n\}$. In more general settings, plans may involve *concurrency* (*i.e.* distribute the available computational power between several simultaneous attacks) and *pauses* (*i.e.* halt an attack at a certain point in time and resume it later). This work does not deal with concurrency and pauses.

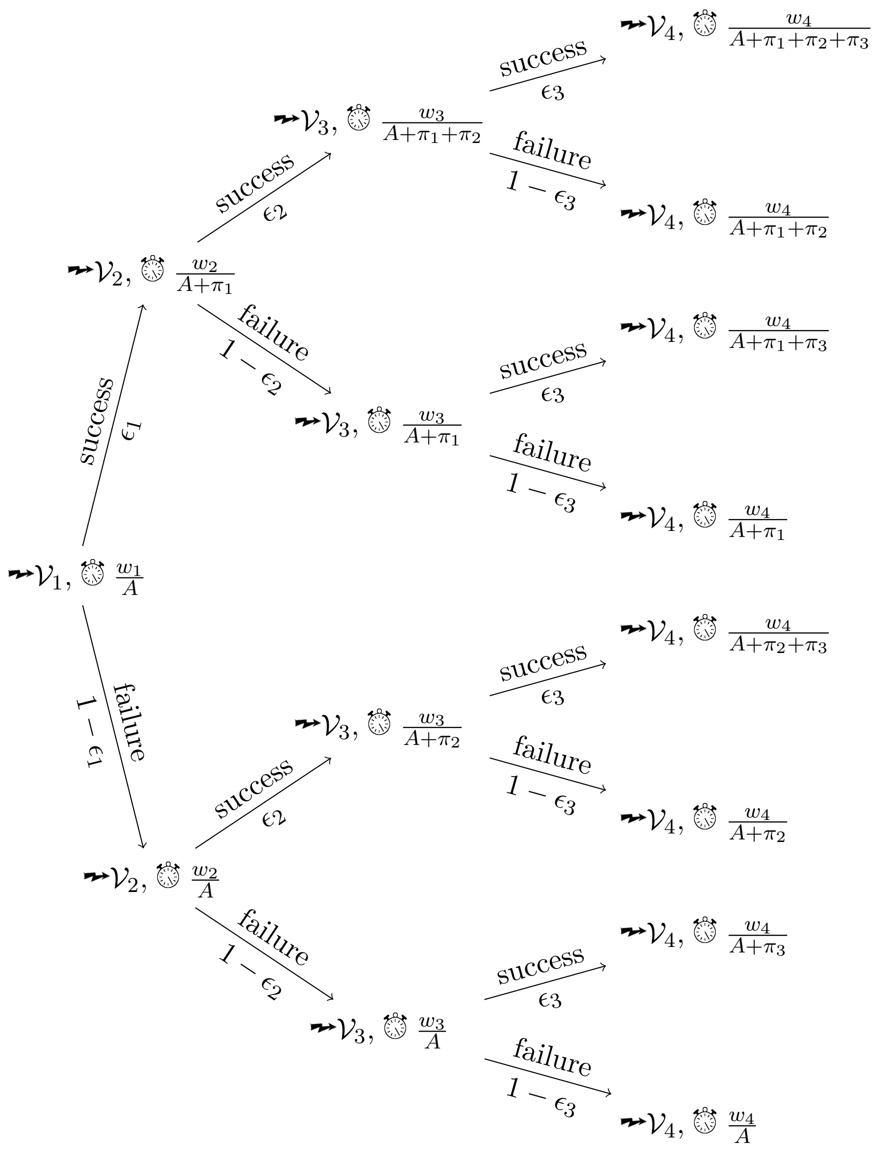
To simplify diagrams, the pictographs $\rightsquigarrow\mathcal{V}_i$, ⏱ t_i will mean "*Alice attacks $\mathcal{V}_i$ and this attack requires t_i time units*". Γ_n will denote the set of $n!$ permutations of $\{1, \ldots, n\}$.

As we will see, even this very simplified model requires involved analysis. Hence, real-life attacks that involve tens of other parameters would require much more complex heuristics.

2 Deterministic Sequential Attacks

In a *deterministic sequential* plan, $P \in \Gamma_n$. Hence, Alice's goal is to find the P_{op} that minimizes $\Delta_{\text{det}}(P)$.

A glance at the small example $\Delta_{\text{det}}(\{1, 2, 3, 4\})$ easily reveals the structure of $\Delta_{\text{det}}(P)$:


$\mathcal{V}_1$, $\frac{w_1}{A}$
success ϵ_1
failure $1-\epsilon_1$
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_1}$
$\mathcal{V}_2$, $\frac{w_2}{A}$
success ϵ_2
failure $1-\epsilon_2$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_1+\pi_2}$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_1}$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_2}$
$\mathcal{V}_3$, $\frac{w_3}{A}$
success ϵ_3
failure $1-\epsilon_3$
$\mathcal{V}_4$, $\frac{w_4}{A+\pi_1+\pi_2+\pi_3}$
$\mathcal{V}_4$, $\frac{w_4}{A+\pi_1+\pi_2}$
$\mathcal{V}_4$, $\frac{w_4}{A+\pi_1+\pi_3}$
$\mathcal{V}_4$, $\frac{w_4}{A+\pi_1}$
$\mathcal{V}_4$, $\frac{w_4}{A+\pi_2+\pi_3}$
$\mathcal{V}_4$, $\frac{w_4}{A+\pi_2}$
$\mathcal{V}_4$, $\frac{w_4}{A+\pi_3}$
$\mathcal{V}_4$, $\frac{w_4}{A}$

and mathematically:

$$
\begin{aligned}
&\Delta_{\text{det}}(\{1,2,3,4\}) = \\
&\quad \epsilon_1\epsilon_2\epsilon_3 \left(\frac{w_1}{A} + \frac{w_2}{A+\pi_1} + \frac{w_3}{A+\pi_1+\pi_2} + \frac{w_4}{A+\pi_1+\pi_2+\pi_3}\right) + \\
&\quad \epsilon_1\epsilon_2(1-\epsilon_3) \left(\frac{w_1}{A} + \frac{w_2}{A+\pi_1} + \frac{w_3}{A+\pi_1+\pi_2} + \frac{w_4}{A+\pi_1+\pi_2}\right) + \\
&\quad \epsilon_1(1-\epsilon_2)\epsilon_3 \left(\frac{w_1}{A} + \frac{w_2}{A+\pi_1} + \frac{w_3}{A+\pi_1} + \frac{w_4}{A+\pi_1+\pi_3}\right) + \\
&\quad \epsilon_1(1-\epsilon_2)(1-\epsilon_3) \left(\frac{w_1}{A} + \frac{w_2}{A+\pi_1} + \frac{w_3}{A+\pi_1} + \frac{w_4}{A+\pi_1}\right) + \\
&\quad (1-\epsilon_1)\epsilon_2\epsilon_3 \left(\frac{w_1}{A} + \frac{w_2}{A} + \frac{w_3}{A+\pi_2} + \frac{w_4}{A+\pi_2+\pi_3}\right) + \\
&\quad (1-\epsilon_1)\epsilon_2(1-\epsilon_3) \left(\frac{w_1}{A} + \frac{w_2}{A} + \frac{w_3}{A+\pi_2} + \frac{w_4}{A+\pi_2}\right) + \\
&\quad (1-\epsilon_1)(1-\epsilon_2)\epsilon_3 \left(\frac{w_1}{A} + \frac{w_2}{A} + \frac{w_3}{A} + \frac{w_4}{A+\pi_3}\right) + \\
&\quad (1-\epsilon_1)(1-\epsilon_2)(1-\epsilon_3) \left(\frac{w_1}{A} + \frac{w_2}{A} + \frac{w_3}{A} + \frac{w_4}{A}\right)
\end{aligned}
$$

For instance, the term:

$$\epsilon_1(1-\epsilon_2)\epsilon_3 \left(\frac{w_1}{A} + \frac{w_2}{A+\pi_1} + \frac{w_3}{A+\pi_1} + \frac{w_4}{A+\pi_1+\pi_3}\right)$$

expresses the fact that when the attack starts, Alice must attack $\mathcal{V}_1$ using her own computer (during $\frac{w_1}{A}$ time units). Because $\mathcal{V}_1$ falls into Alice's hands (ϵ_1), the attack on $\mathcal{V}_2$ uses both Alice's own means A and $\mathcal{V}_1$'s captured power π_1. $\frac{w_2}{A+\pi_1}$ time units are hence required to attack $\mathcal{V}_2$ but this attack fails $(1-\epsilon_2)$. Hence $\mathcal{V}_3$ is attacked using $A+\pi_1$ only during $\frac{w_3}{A+\pi_1}$ time. $\mathcal{V}_3$ falls (ϵ_3) and Alice's power grows further to $A+\pi_1+\pi_3$ which allows her to complete the (last) attack on $\mathcal{V}_4$ in $\frac{w_4}{A+\pi_1+\pi_3}$ time units. In other words, the formula enumerates all success/failure scenarios and sums their respective time expectations.

For arbitrary n, Δ_{det} is given by the following formula where $k[j]$ denotes j-th bit of the integer k:

$$\Delta_{\text{det}}(P) = \sum_{k=0}^{2^{n-1}-1} \prod_{j=0}^{n-2} \epsilon_{P(j+1)}^{k[j]} (1-\epsilon_{P(j+1)})^{1-k[j]} \sum_{i=1}^{n} \frac{w_{P(i)}}{A + \sum\limits_{\ell=0}^{i-2} k[\ell]\pi_{P(\ell+1)}}$$

The evolution of Δ_{det} is irregular because $\Delta_{\text{det}}(P)$ does not depend on $\pi_{P(n)}$ and $\epsilon_{P(n)}$. Thus, swapping $\mathcal{V}_{P(n)}$ with some another $\mathcal{V}_{i\neq P(n)}$ replaces two formula parameters by new ones. This may cause radical variations in Δ.

2.1 Particular cases

Fixed powers Let's assume all π_i are equal to p, then by sorting all computers according to $f(w, \varepsilon) = \frac{\varepsilon}{w}$ in decreasing order, we obtain the optimal solution. The complexity of this algorithm is obviously the complexity of an efficient sort, that is $O(n \log n)$.

Proof: We will suppose that the list of computer is sorted in decreasing order .

Let σ be a permutation different from identity.

let $i = \min(1 \leq j \leq n, \sigma(j) > \sigma(j+1))$ i exist because σ is not identity.

Then let ω such as $\omega = \sigma \circ (i \;\; i+1)$ where $(i \;\; i+1)$ represent the permutation of i and $i+1$

Let $T(\sigma)$ the average time of the attack in the order of σ and $T(\omega)$ the time of the attack with ω

The only difference between trees of σ and ω are branches at depth i and $i+1$ Then in all branches we have if A is the power at depth i of this branch ,$u = \sigma(i)$ and $v = \sigma(i+1)$: $T(\sigma) - T(\omega) = \frac{w_u}{A} + \epsilon_u(\frac{w_v}{A+1} + (1-\epsilon_u)(\frac{w_v}{A}) - \frac{w_v}{A} + \epsilon_v(\frac{w_u}{A+1}) + (1-\epsilon_v)(\frac{w_u}{A} = (\frac{-1}{A(A+1)})(\epsilon_u w_v - \epsilon_v w_u = w_u w_v(\frac{1}{A(A+1)})(\frac{\epsilon_v}{w_v} - \frac{\epsilon_u}{w_u}) \geq 0$ because $v < u$

So any order exept the decreasing one can be enhanced .Finally The decreasing order is the best.

2.2 Computational Strategies

We did not find polynomial-time algorithms for computing P_{op}. However, as we will immediately see, the problem's complexity is (at most) exponential and not factorial.

We first note that the computation of P_{op} can be accelerated (by a constant factor) using early aborts[5] and by smartly recycling for P_{i+1} computations performed for computing P_i.

Also, if $P = \{P(1), \ldots, P(n-1), P(n)\}$ and $P' = \{P(1), \ldots, P(n-2), P(n), P(n-1)\}$ we observe that the common prefix $\{P(1), \ldots, P(n-2)\}$ can serve for the computation of both $\Delta_{\text{det}}(P)$ and $\Delta_{\text{det}}(P')$. Pushing this observation further, we get a divide and conquer algorithm:

Let $n = 2k$ and let P^{2k}_{op} be the optimal plan for attacking $\mathcal{V}_1, \ldots, \mathcal{V}_{2k}$. Given the (unordered) list $\mathcal{V}_{P^{2k}_{\text{op}}(1)}, \ldots, \mathcal{V}_{P^{2k}_{\text{op}}(k)}$ we can compute the (ordered) prefix $\{P^{2k}_{\text{op}}(1), \ldots, P^{2k}_{\text{op}}(k)\}$ of P^{2k}_{op} without examining $\mathcal{V}_{P^{2k}_{\text{op}}(k+1)}, \ldots, \mathcal{V}_{P^{2k}_{\text{op}}(2k)}$, *i.e.* $\Delta_{\text{det}}(P^{2k}_{\text{op}})$ can be computed by generating and solving all $\binom{2k}{k}$ sub-problem pairs of size k.

We begin with the following formula :

$$T(2n, r) = \frac{2n!}{n!^2}(T(n, r) + T(n, r+n))$$

$$T(2n+1, r) = \frac{2(n+1)!}{(n+1)!n!}(T(n+1, r) + T(n, r+n+1))$$

$$T(1, r) = 2^r$$

[5] *e.g.* start by recording $\Delta_{\text{det}}(P_0)$ as a best score and begin computing $\Delta_{\text{det}}(P_1)$. As soon as the summation of terms in $\Delta_{\text{det}}(P_1)$ exceeds $\Delta_{\text{det}}(P_0)$ stop and try P_2 etc

We assume $\exists K > 0, \forall n \geqslant 16, \forall r \in \mathbb{N}, T(n,r) \leqslant K(2^{3n+r})$, thereby :

$$T(2n,r) \leqslant \frac{4^n}{\sqrt{n\pi}} K(2^{3n+r} + 2^{3n+n+r})$$

also

$$T(2n+1,r) \leqslant \frac{4^n}{\sqrt{n\pi}} K(2^{3(n+1)+r} + 2^{3n+n+r+1})$$

thus

$$T(2n,r) \leqslant \frac{2}{\sqrt{n\pi}} K 2^{3(2n)+r} \leqslant K 2^{3(2n)+r}$$

$$T(2n+1,r) \leqslant \frac{4}{\sqrt{n\pi}} K 2^{3(2n)+r} \leqslant K 2^{3(2n)+r}$$

Furthermore it remains true for $n \leqslant 15$, thus by induction :

$$\exists K > 0, \forall n \in \mathbb{N}, \forall r \in \mathbb{N}, T(n,r) \leqslant K 2^{3n+r}$$

In other words, if the cost of finding P_{op}^{2k} is $O(2^{3n})$.

2.3 Heuristics

Algorithms A simple heuristic is to order the computers by a function of w, π and ε and possibly A. For example one can easily intuit $f(w,\pi,\varepsilon) = \frac{\pi\varepsilon}{w}$ to be interesting. This function indeed yields good results for small instance, but one can do better. Let us consider one instance of the problem and two computers $(w_i, \pi_i, \varepsilon_i)$ and $(w_j, \pi_j, \varepsilon_j)$. We can decide easily if it is better to attack one computer before immediately attacking the other.

$$\begin{aligned}
\Delta(\{i,j,\cdots\}) - \Delta(\{j,i,\cdots\}) &= \frac{w_i}{A} + \varepsilon_i \frac{w_j}{A+\pi_i} + (1-\varepsilon_i)\frac{w_j}{A} - \left(\frac{w_j}{A} + \varepsilon_j \frac{w_i}{A+\pi_j} + (1-\varepsilon_j)\frac{w_i}{A}\right) \\
&= \varepsilon_i w_j \left(\frac{1}{A+\pi_i} - \frac{1}{A}\right) - \varepsilon_j w_i \left(\frac{1}{A+\pi_j} - \frac{1}{A}\right) \\
&= -\varepsilon_i w_j \frac{\pi_i}{A(A+\pi_i)} + \varepsilon_j w_i \frac{\pi_j}{A(A+\pi_j)}
\end{aligned}$$

Thus it follows that :

$$\Delta(\{i,j,\cdots\}) < \Delta(\{j,i,\cdots\}) \Leftrightarrow \varepsilon_i w_j \frac{\pi_i}{A+\pi_i} > \varepsilon_j w_i \frac{\pi_j}{A+\pi_j} \Leftrightarrow \frac{\varepsilon_i \pi_i}{w_i(A+\pi_i)} > \frac{\varepsilon_j \pi_j}{w_j(A+\pi_j)}$$

We can now order the computers according to the values (in decreasing order) of the function $f(A,w,\pi,\varepsilon) = \frac{\varepsilon\pi}{w(A+\pi)}$. Of course after the first step, we no longer have a fixed power, although we could take the mean of the values of f on different branches. However keeping the initial power gives good approximation with the complexity of a sort. This heuristics works well because we approximate a permutation of the computer which, albeit potentially suboptimal, is close to a permutation that is not improvable by any transposition of the form $(i \;\; i+1)$

Actually it is possible to obtain a permutation of the computers that can't be improved by any transposition of the form $(i \;\; i+1)$, at the price of an increased complexity of $O(2^n)$. This highlights the difficulty of the problem, as it is possible to find instances where such an algorithm is stuck in a suboptimal solution.

A simple way to achieve an approximation of the last algorithm, which is polynomial in time, is to take the mean across sampled paths.

Performance

3 An Experimental Observation

Can a P_{op}^{n+1} be constructed by inserting $\mathcal{V}_{n+1}$ into P_{op}^{n}? It appears that such is frequently the case. This section provides experimental statistics about the ratio of configurations for which the insertion of $\mathcal{V}_5$ and $\mathcal{V}_6$ into a 4-target solution is possible.

We performed the following experiment

```
proc Experiment($B_\pi, B_w$) $\equiv$
  for $A := 1$ to 400 do
    $r_A := 0$
    for $t := 1$ to 800 do
      for $i := 1$ to 6 do
        generate randomly $0.01 \leq \epsilon_i \leq 1$
        generate randomly $1 \leq \pi_i \leq B_\pi$
        generate randomly $1 \leq w_i \leq B_w$
      od
      compute $P_{\text{op}}(\mathcal{V}_1, \ldots, \mathcal{V}_6)$
      compute $P_{\text{op}}(\mathcal{V}_1, \ldots, \mathcal{V}_4)$
      if $P_{\text{op}}(\mathcal{V}_1, \ldots, \mathcal{V}_6)$ can be obtained by inserting 5 and 6 into $P_{\text{op}}(\mathcal{V}_1, \ldots, \mathcal{V}_4)$
        then $r_A := r_A + \frac{1}{800}$
      fi
    od
  od
```

It appears that for large A the success ratio r_A tends to one. This phenomenon is arguably related to the fact that sorting the computers according

4 Adaptive Attacks

In an adaptive attack, Alice adapts future moves to successes and failures. To understand what an optimal adaptive attack is, assume that we already know (thanks to some oracle) that the best adaptive attack *must* start by attacking $\mathcal{V}_1$. Before attacking $\mathcal{V}_1$ we face the two possible futures illustrated in Figures 9 and 7:

After attacking $\mathcal{V}_1$ the "dust settles" and Alice's information increases. Hence:

$$\begin{aligned}\Delta_{\mathsf{ad}}(\{1, \bullet, \bullet\}) = {} & \frac{w_1}{A} + \\ \epsilon_1 \min & \left(\frac{w_2}{A+\pi_1} + \frac{\epsilon_2 w_3}{A+\pi_1+\pi_2} + \frac{(1-\epsilon_2)w_3}{A+\pi_1}, \frac{w_3}{A+\pi_1} + \frac{\epsilon_3 w_2}{A+\pi_1+\pi_3} + \frac{(1-\epsilon_3)w_2}{A+\pi_1} \right) + \\ (1-\epsilon_1) \min & \left(\frac{w_2}{A} + \frac{\epsilon_2 w_3}{A+\pi_2} + \frac{(1-\epsilon_2)w_3}{A}, \frac{w_3}{A} + \frac{\epsilon_3 w_2}{A+\pi_3} + \frac{(1-\epsilon_3)w_2}{A} \right)\end{aligned}$$

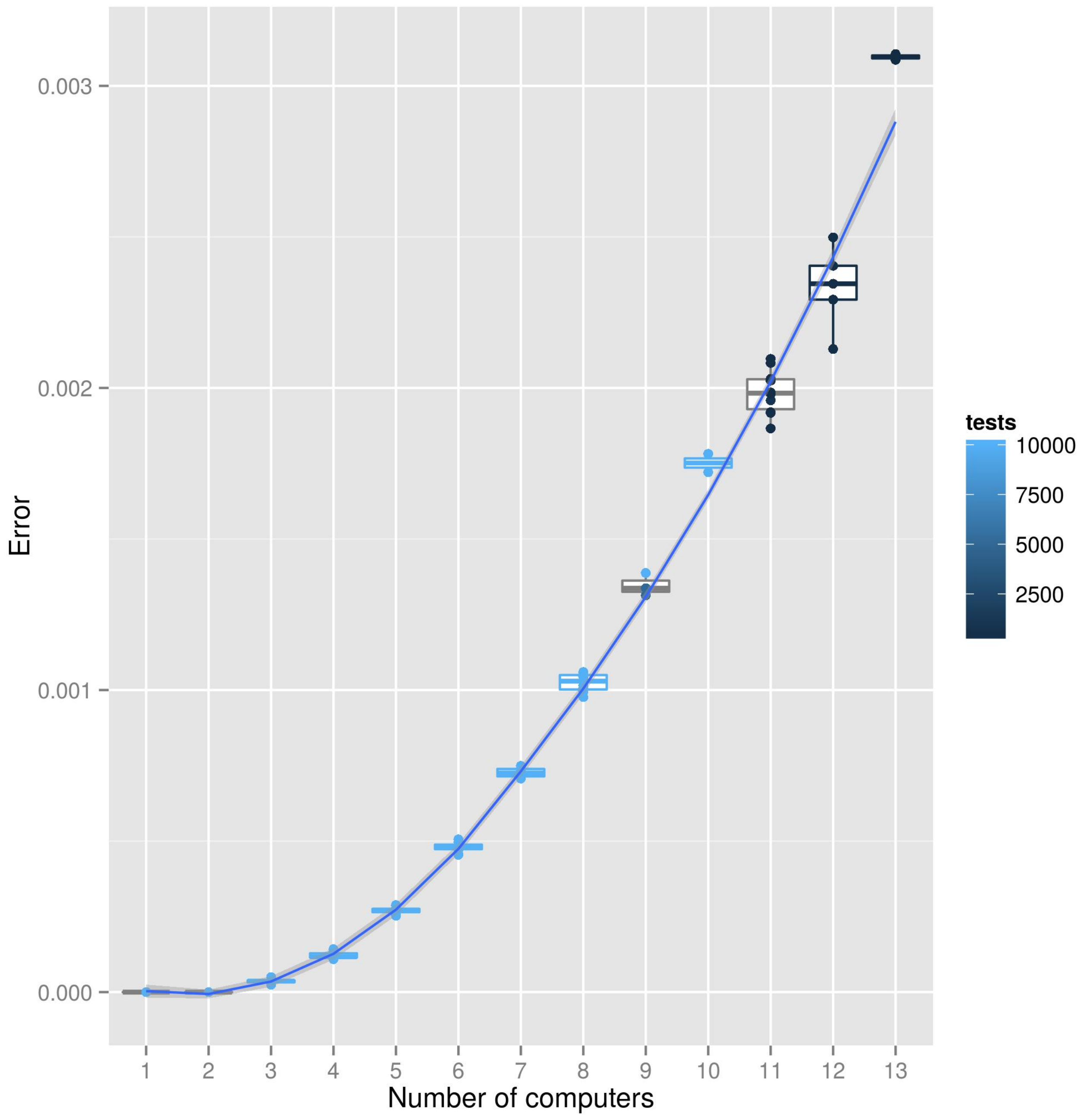


Fig. 2. Error of the heuristic compared to the optimal

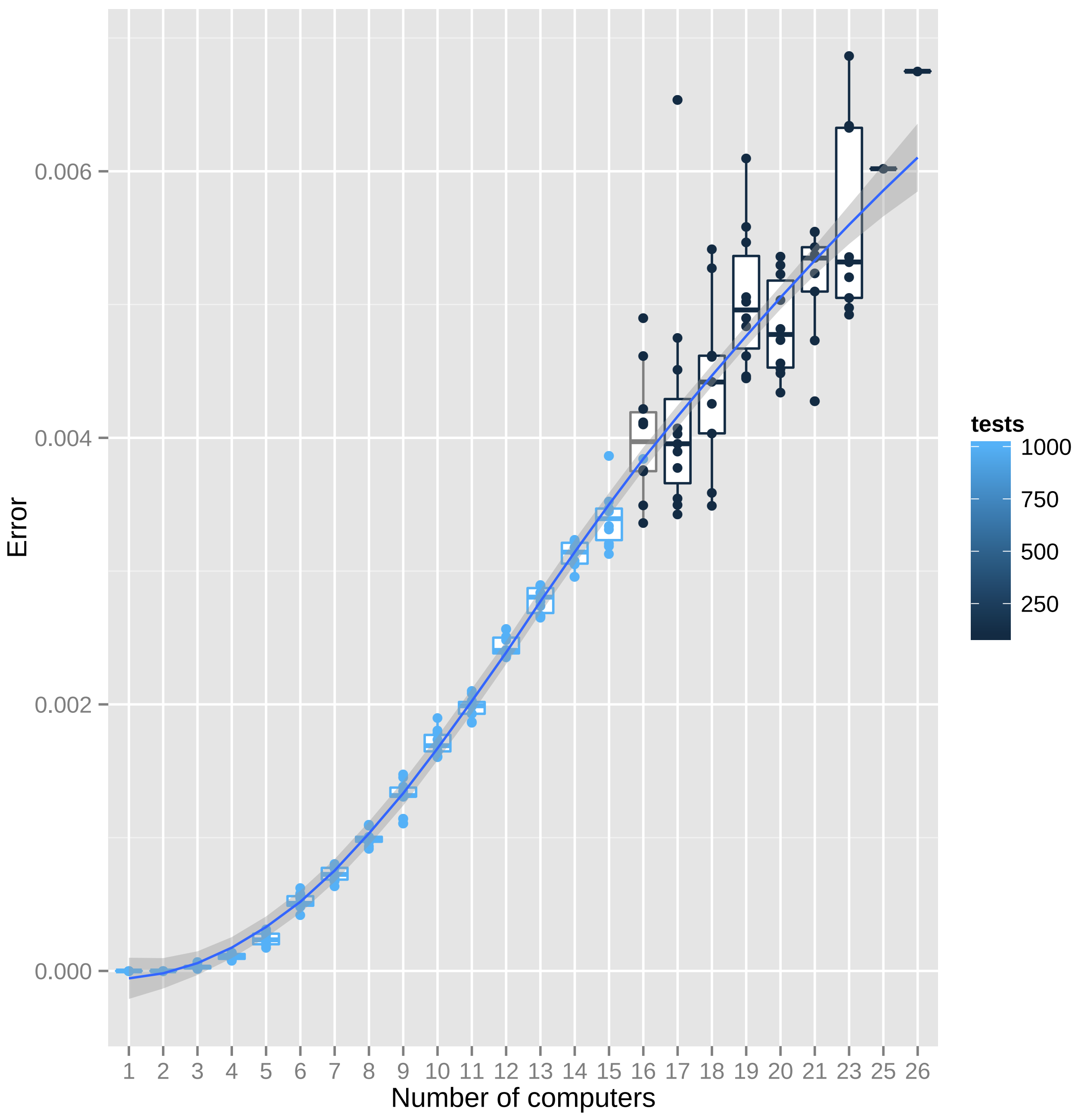


Fig. 3. Error of the heuristic compared to greedily ordering pairs.

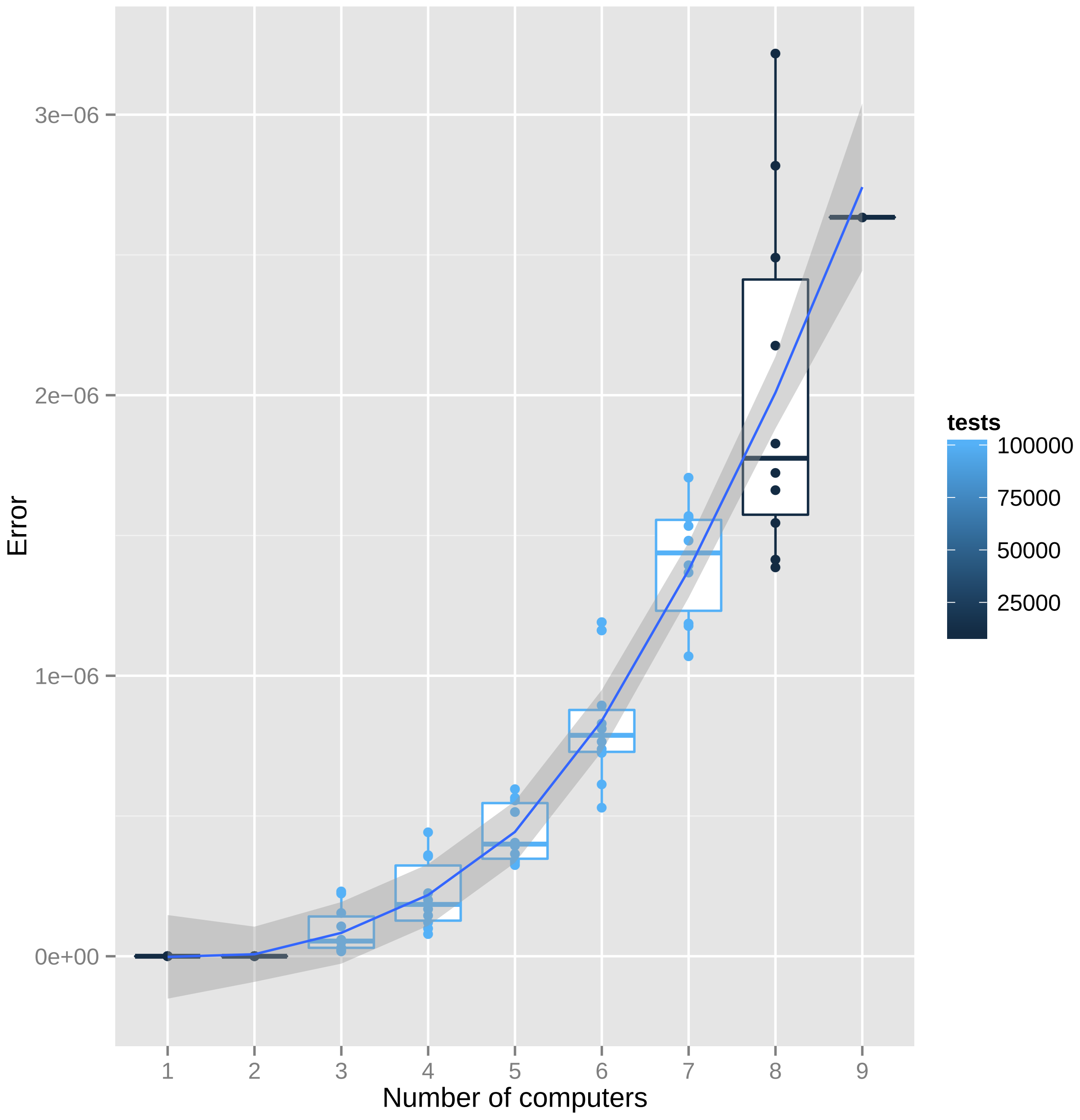


Fig. 4. Error of the the greedy algorithm compared to the optimal.

Fig. 5. Experiment$(100, 100)$ (in red) and Experiment$(50, 200)$ (in blue). r_A is the ratio of 6-target optimal plans obtainable by inserting 5 and 6 into their corresponding 4-target optimal plan.

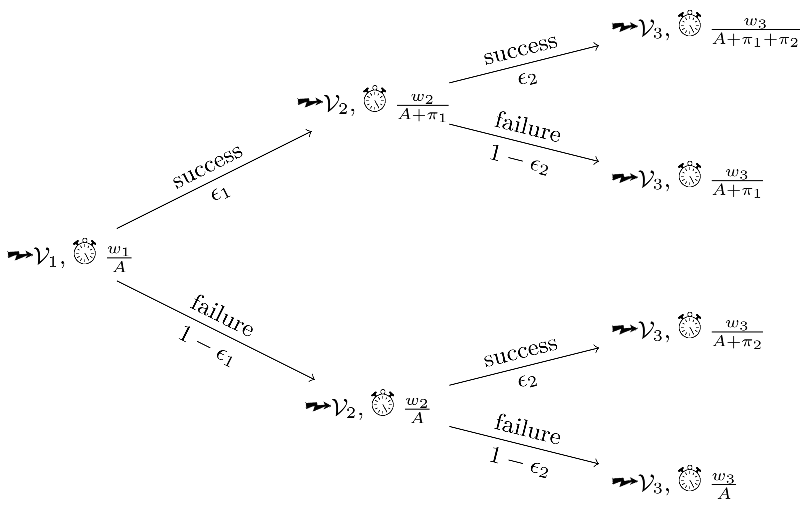

$\mathcal{V}_1$, $\frac{w_1}{A}$
success
ϵ_1
failure
$1-\epsilon_1$
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_1}$
$\mathcal{V}_2$, $\frac{w_2}{A}$
success
ϵ_2
failure
$1-\epsilon_2$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_1+\pi_2}$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_1}$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_2}$
$\mathcal{V}_3$, $\frac{w_3}{A}$

Fig. 6. First possible future after attacking $\mathcal{V}_1$: attack $\mathcal{V}_2$ first, $\mathcal{V}_3$ next.

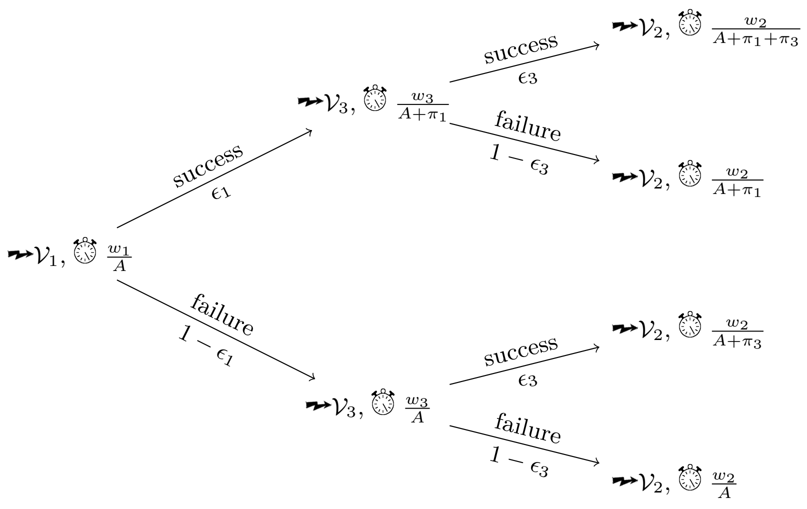

$\mathcal{V}_1$, $\frac{w_1}{A}$
success
ϵ_1
failure
$1-\epsilon_1$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_1}$
$\mathcal{V}_3$, $\frac{w_3}{A}$
success
ϵ_3
failure
$1-\epsilon_3$
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_1+\pi_3}$
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_1}$
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_3}$
$\mathcal{V}_2$, $\frac{w_2}{A}$

Fig. 7. Second possible future after attacking $\mathcal{V}_1$: attack $\mathcal{V}_3$ first, $\mathcal{V}_2$ next.

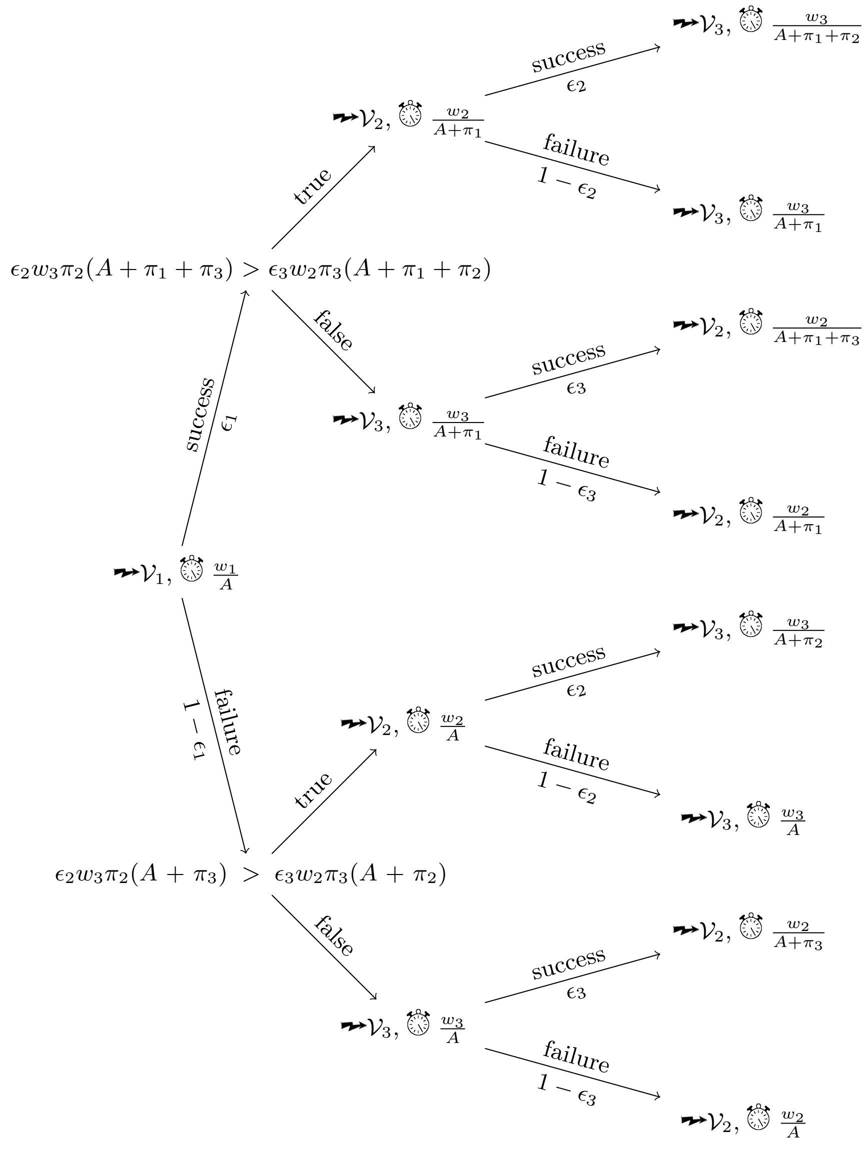
It appears that

$$\frac{w_2}{A+\pi_1} + \frac{\epsilon_2 w_3}{A+\pi_1+\pi_2} + \frac{(1-\epsilon_2)w_3}{A+\pi_1} < \frac{w_3}{A+\pi_1} + \frac{\epsilon_3 w_2}{A+\pi_1+\pi_3} + \frac{(1-\epsilon_3)w_2}{A+\pi_1}$$
$$\Downarrow$$
$$\epsilon_2 w_3 \pi_2 (A+\pi_1+\pi_3) > \epsilon_3 w_2 \pi_3 (A+\pi_1+\pi_2)$$

and

$$\frac{w_2}{A} + \frac{\epsilon_2 w_3}{A+\pi_2} + \frac{(1-\epsilon_2)w_3}{A} < \frac{w_3}{A} + \frac{\epsilon_3 w_2}{A+\pi_3} + \frac{(1-\epsilon_3)w_2}{A}$$
$$\Downarrow$$
$$\epsilon_2 w_3 \pi_2 (A+\pi_3) > \epsilon_3 w_2 \pi_3 (A+\pi_2)$$

Figures 9 and 7 can thus be merged into the unique plan shown in Figure 8.



$\mathcal{V}_1$, $\frac{w_1}{A}$
success ϵ_1
failure $1-\epsilon_1$
$\epsilon_2 w_3 \pi_2 (A+\pi_1+\pi_3) > \epsilon_3 w_2 \pi_3 (A+\pi_1+\pi_2)$
true
false
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_1}$
success ϵ_2
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_1+\pi_2}$
failure $1-\epsilon_2$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_1}$
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_1}$
success ϵ_3
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_1+\pi_3}$
failure $1-\epsilon_3$
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_1}$
$\epsilon_2 w_3 \pi_2 (A+\pi_3) > \epsilon_3 w_2 \pi_3 (A+\pi_2)$
true
false
$\mathcal{V}_2$, $\frac{w_2}{A}$
success ϵ_2
$\mathcal{V}_3$, $\frac{w_3}{A+\pi_2}$
failure $1-\epsilon_2$
$\mathcal{V}_3$, $\frac{w_3}{A}$
$\mathcal{V}_3$, $\frac{w_3}{A}$
success ϵ_3
$\mathcal{V}_2$, $\frac{w_2}{A+\pi_3}$
failure $1-\epsilon_3$
$\mathcal{V}_2$, $\frac{w_2}{A}$

Fig. 8. Combined attack plan starting with $\mathcal{V}_1$.

To find the best adaptive attack, Alice defines:

$$\begin{aligned}\mathit{\Delta}_{\mathsf{ad}}(\{x,\bullet,\bullet\}) = & \frac{w_x}{A} + \\ (1-\epsilon_x)\min & \left(\frac{w_z}{A}+\frac{\epsilon_z w_y}{A+\pi_z}+\frac{(1-\epsilon_z)w_y}{A}, \frac{w_y}{A}+\frac{\epsilon_y w_z}{A+\pi_y}+\frac{(1-\epsilon_y)w_z}{A}\right) \\ \epsilon_x \min & \left(\frac{w_z}{A+\pi_x}+\frac{\epsilon_z w_y}{A+\pi_x+\pi_z}+\frac{(1-\epsilon_z)w_y}{A+\pi_x}, \frac{w_y}{A+\pi_x}+\frac{\epsilon_y w_z}{A+\pi_x+\pi_y}+\frac{(1-\epsilon_y)w_z}{A+\pi_x}\right)\end{aligned}$$

and computes:

$$\mathit{\Delta}_{\mathsf{ad}}(P_{\mathsf{opt}}) = \min_{\{x,y,z\}\in\mathit{\Gamma}_n}(\mathit{\Delta}_{\mathsf{ad}}(x,\bullet,\bullet))$$

Intuition suggests that an adaptive plan (allowing Alice more freedom of action) would yield better results than a deterministic one. Such is indeed the case, as shown in the following example[6] where $A = 7$, $n = 3$ and $\mathcal{V}_1 = \{2, 1, 0.3\}$, $\mathcal{V}_2 = \{10, 5, 0.2\}$, $\mathcal{V}_3 = \{1, 2, 0.4\}$ and:

permutation $\rightarrow$	$\{1,2,3\}$	$\{1,3,2\}$	$\{2,1,3\}$	$\{2,3,1\}$	$\{3,1,2\}$	$\{3,2,1\}$
$\mathit{\Delta}_{\mathsf{det}}$	1.04564	1.04452	1.07646	1.08646	1.05643	1.08436

For $n = 3$ adaptive attacks are not characterized by a permutation but by the $\mathcal{V}_i$ attacked first. It appears that:

$$\begin{aligned}\mathit{\Delta}_{\mathsf{det}}(2,1,3) = & \mathit{\Delta}_{\mathsf{ad}}(2,\bullet,\bullet) = 1.07646 \\ \mathit{\Delta}_{\mathsf{det}}(3,1,2) = & \mathit{\Delta}_{\mathsf{ad}}(3,\bullet,\bullet) = 1.05643 \\ \text{but:} \quad & \mathit{\Delta}_{\mathsf{ad}}(1,\bullet,\bullet) = 1.04417 < \min_{\{x,y,z\}\in\mathit{\Gamma}_3}(\mathit{\Delta}_{\mathsf{det}}(x,y,z)) = \mathit{\Delta}_{\mathsf{det}}(1,3,2) = 1.04452\end{aligned}$$

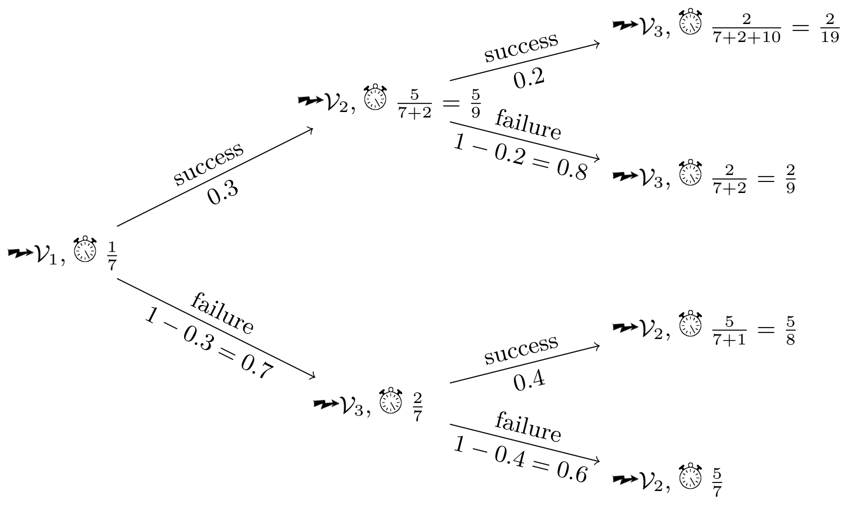
$$\begin{aligned}\text{Here:} \quad \mathit{\Delta}_{\mathsf{ad}}(1,\bullet,\bullet) = \frac{w_1}{A} + (1-\epsilon_1)\left(\frac{(1-\epsilon_3)w_2}{A}+\frac{\epsilon_3 w_2}{A+\pi_3}+\frac{w_3}{A}\right) + \\ \epsilon_1\left(\frac{w_2}{A+\pi_1}+\frac{(1-\epsilon_2)w_3}{A+\pi_1}+\frac{\epsilon_2 w_3}{A+\pi_1+\pi_2}\right) = 1.04417\end{aligned}$$

We have

$$\frac{\varepsilon_2\pi_2}{w_2}(A+\pi_1+\pi_3) > \frac{\varepsilon_3\pi_3}{w_3}(A+\pi_1+\pi_2) \qquad (1)$$

$$\frac{\varepsilon_2\pi_2}{w_2}(A+\pi_3) < \frac{\varepsilon_3\pi_3}{w_3}(A+\pi_2) \qquad (2)$$

[6] $\mathcal{V}_i = \{\pi_i, w_i, \epsilon_i\}$



$\mathcal{V}_1$, $\frac{1}{7}$
success
0.3
failure
$1-0.3=0.7$
$\mathcal{V}_2$, $\frac{5}{7+2}=\frac{5}{9}$
$\mathcal{V}_3$, $\frac{2}{7}$
success
0.2
failure
$1-0.2=0.8$
$\mathcal{V}_3$, $\frac{2}{7+2+10}=\frac{2}{19}$
$\mathcal{V}_3$, $\frac{2}{7+2}=\frac{2}{9}$
success
0.4
failure
$1-0.4=0.6$
$\mathcal{V}_2$, $\frac{5}{7+1}=\frac{5}{8}$
$\mathcal{V}_2$, $\frac{5}{7}$

Fig. 9. The 1, ●, ● scenario (note that the order of attacks in the branches is reversed).

It follows that :

$$\begin{aligned}
-\frac{\varepsilon_2\pi_2}{w_2}(A+\pi_3) &> -\frac{\varepsilon_3\pi_3}{w_3}(A+\pi_2)\\
\frac{\varepsilon_2\pi_2}{w_2}\pi_1 &> \frac{\varepsilon_3\pi_3}{w_3}\pi_1\\
\frac{\varepsilon_2\pi_2}{w_2} &> \frac{\varepsilon_3\pi_3}{w_3}\\
-A\frac{\varepsilon_2\pi_2}{w_2} &< -A\frac{\varepsilon_3\pi_3}{w_3}\\
\frac{\varepsilon_2\pi_2}{w_2}\pi_3 &< \frac{\varepsilon_3\pi_3}{w_3}\pi_2
\end{aligned}$$

Finally :

$$\frac{\varepsilon_2}{w_2} > \frac{\varepsilon_3}{w_3} \tag{3}$$

$$\frac{\varepsilon_2\pi_2}{w_2} < \frac{\varepsilon_3\pi_3}{w_3} \tag{4}$$

We can sum up :

$$\frac{\varepsilon_2\pi_2}{w_2\pi_3} > \frac{\varepsilon_3}{w_3} > \frac{\varepsilon_2}{w_2}$$

There is no dependency on π_1 because when only two computers are left, the current power The general algorithm is illustrated in Figures 10 and 1. Assume again that we know (thanks to some oracle) that an adaptive attack for $n=4$ should start by $\mathcal{V}_1$. Figure 1 shows the 4 possible attack plans starting by an attack on $\mathcal{V}_1$.

To compute a time expectation of a branch, consider the chain of red and blue cells leading to the black leaf. The chain defines the computation power available for attacking the black leaf and hence the time taken to do so. Work the way up until a ● is met. Then prune all branches except the one whose time is minimal and proceed further up. This will yield a decision tree representing the optimal adaptive moves.

Now, because we are not given an oracle (*i.e.* the optimal attack may begin by some $\mathcal{V}_i \neq \mathcal{V}_1$) the process must be repeated for all possible first targets as shown in Figure 10.

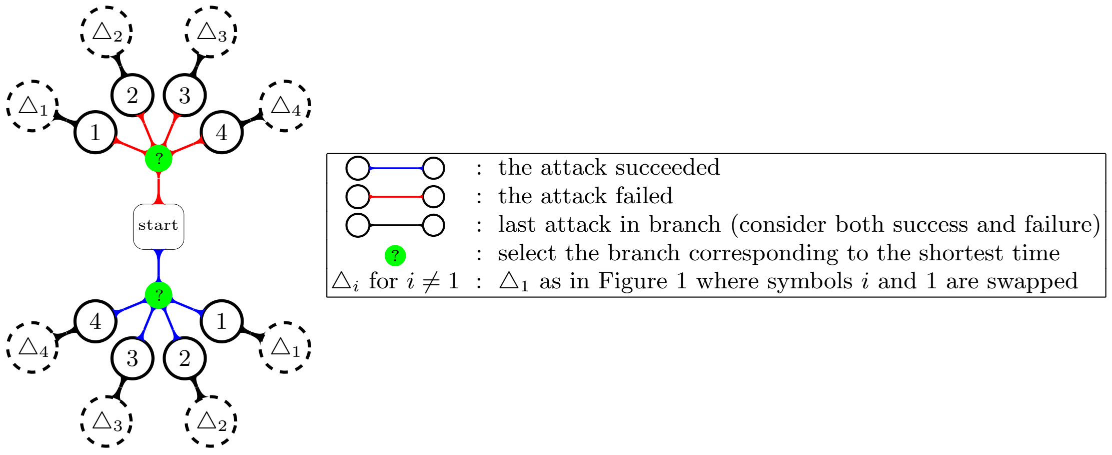
The algorithm examines $2^n n!$ attack chains and prunes them to get the 2^n "recipe" describing how to best proceed adaptively.

Here are the frequencies of different timings:

2	{4,1}, {2,2}
3	{12,1}, {2,12}, {4,3}
4	{48,1}, {4,36}, {2,72}, {12,4}
5	{240,1}, {12,60}, {8,30}, {4,300}, {2,600}, {48,5}
6	{1440,1}, {48,90}, {24,120}, {12,600}, {4,3600}, {8,360}, {2,5760}, {240,6}
7	{10080,1}, {240,126}, {96,210}, {48,1050}, {72,140}, {12,8400}, {24,1680}, {8,7560}, {4,45360}, {2,65520}, {1440,7}

We could heuristically identify some of these entries as:

$$\{2i!, 1\},\ \{2, i!\phi(i)\}, \{2(i-1)!, i\}, \{2(i-2)!, 3i(i-1)\}$$



start
?
1
2
3
4
$\triangle_1$
$\triangle_2$
$\triangle_3$
$\triangle_4$
: the attack succeeded
: the attack failed
: last attack in branch (consider both success and failure)
: select the branch corresponding to the shortest time
$\triangle_i$ for $i \neq 1$: $\triangle_1$ as in Figure 1 where symbols i and 1 are swapped

Fig. 10. Finding an adaptive P_{opt} for $n = 4$ (start of process).

References

1. http://searchsecurity.techtarget.com/definition/botnet
2. M. Abu Rajab, J. Zarfoss, F. Monrose, A. Terzis, A multifaceted approach to understanding the botnet phenomenon, in Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC 06)