Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

Near-optimal blacklisting

Christos Dimitrakakis, Aikaterini Mitrokotsa *

Chalmers University of Technology, Gothenburg, Sweden

ARTICLE INFO

Article history:
Received 21 August 2014
Received in revised form 25 April 2015
Accepted 26 June 2015
Available online 8 July 2015

Keywords:
Decision theory
Blacklisting
Markov decision process
Optimal stopping
Expected loss
Network management

ABSTRACT

Many applications involve agents sharing a resource, such as networks or services. When agents are honest, the system functions well and there is a net profit. Unfortunately, some agents may be malicious, but it may be hard to detect them. We consider the *decision making* problem of how to permanently *blacklist* agents, in order to maximise expected profit. The problem of efficiently deciding which nodes to permanently blacklist has various applications ranging from efficient intrusion response, network management, shutting down malware infected hosts in an internal network and efficient distribution of services in a network. In this paper, we propose an approach to efficiently perform this blacklisting while minimising the cost of the service provider. Although our approach is quite general and could be applied to all the previously mentioned applications, to ease understanding we consider the problem in which an Internet service provider (ISP) needs to decide whether or not to blacklist a possibly misbehaving node. This is not trivial, as blacklisting may erroneously expel honest nodes (agents). Conversely, while we gain information by allowing a node to remain, we may incur a cost due to malicious behaviour. We present an efficient algorithm (HIPER) for making near-optimal decisions for this problem. Additionally, we derive three algorithms by reducing the problem to a Markov decision process (MDP). Theoretically, we show that HIPER is near-optimal. Experimentally, its performance is close to that of the full MDP solution, when the (stronger) requirements of the latter are met.

1. Introduction

We consider the *decision making* problem of blacklisting potentially malicious nodes or agents that share a resource or network based on partial information. As motivation, consider a communication network which is monitored by a network management system employed for example by an Internet service provider (ISP). All nodes pay to participate in the network (i.e. fees to the ISP). However, nodes can be of one of two types: malicious (e.g. sending spam emails, creating undue congestion as part of a denial of service attack, participating in phishing) or honest. For instance, the malicious nodes might be compromised and part of a botnet; thus, their actions disrupt other clients and might pose a threat to the ISP and its reputation.

The malicious nodes thus incur a cost due to their participation, which is not possible to measure directly. However, at each time-step (e.g. reporting period), the administrator/ISP gets a set of measurements, giving some information about the behaviour of each node during that period. These measurements could be alarms from an intrusion detection system monitoring for instance clients connected to malicious adresses. The decision problem at each time step is whether to blacklist a node, or maintain it in the system (i.e. clients of the ISP) for one more time-step.

We consider that for every honest node in the network, we have some fixed tangible gain at each time period. This would

* *Corresponding author.* Tel.: +46 31 772 10 40
E-mail addresses: chrdimi@chalmers.se (C. Dimitrakakis), aikmitr@chalmers.se (A. Mitrokotsa).
http://dx.doi.org/10.1016/j.cose.2015.06.010
0167-4048/

be the case if all participation was done through a subscription model as in ISPs. On the other hand, we incur a (hidden) cost for each malicious node that participates (i.e. maintenance cost for each node in the network, impact on the reputation of an ISP, disruption of the network). Thus, it is in our interests to kick out malicious nodes as soon as possible, but never to expel honest ones. More precisely, in the ISP setting we want the ISP to keep providing fair services to nodes (clients) that respect the service policies of the ISP and we want to blacklist and expel nodes that misbehave consistently.

We should emphasise that this is *not* an intrusion detection problem. In fact, the readings that we obtain for each node could be seen as the output of some intrusion detection system (IDS) that we consider that the service provider/administrator has at his disposal. Rather, we are more concerned about the decision making aspect as a response to a misbehaving, "malicious" node: *what is the optimal response to the IDS outputs, given assumptions about the cost of malicious behaviour?*

This setting of keeping suspicious nodes in the network until we become more certain about their type appears in many applications such as: (i) blacklisting clients of an ISP, (ii) shutting down malware-infected hosts in an internal network, and (iii) expelling selfish nodes from a peer-to-peer network. In all of the above cases, any single piece of information is not enough to condemn a node to blacklisting. Rather, a sufficient amount of statistics has to be collected before we are sure that removing a node is more beneficial than keeping it. In this paper, we propose and consider a number of algorithms for tackling this problem in a general setting.

More precisely, in our setting, the nodes can be one of two types: honest or malicious. However, we initially start out without knowing what type each node is. Consequently, we must gather data (observations) to reduce our uncertainty about their types. Unfortunately, we can only do so while a node remains within the network (e.g. receive normally the services of an ISP). However, the longer we maintain a malicious node in the network, the more loss we incur. Conversely, once we remove an honest node, we will obtain no more profit from it. So, the problem can be reduced to deciding at what time, or under which conditions, to remove a given node from the network, if at all. Thus, our scenario can be seen as a type of *optimal stopping* problem.

Optimal stopping problems can be modelled as a Markov decision process (DeGroot, 1970) (MDP) in a Bayesian setting, where the states equal the set of information states of the problem. While good approximate solutions to general MDPs are computationally demanding, the algorithm we propose is simple, provably efficient, requires fewer assumptions, and has similar or better performance than MDP approximations.

In this paper, we demonstrate how difficult it is to perform efficient decision making in realistic conditions and problems such as efficient intrusion response and effective network management. Although the use of MDP for solving the intrusion response has been identified before, the existing methods approach the problem in a simplistic way. More precisely, in most existing works it is assumed that the administrator has access to the normally hidden awards and thus that he/she is able to see if he/she gets a cost or gain for each decision that he/she takes. Our paper is the first one that addresses this in a more realistic manner. More precisely, we describe how the problem of blacklisting could be modeled as a *hidden reward stopping* problem.

We treat a sub-class of this problem in which some prior knowledge is given to administrators (ISP or in general a decision-maker). This prior knowledge includes that there are two types of behavior (normal/malicious) and that for each node some information is given by an existing monitoring mechanism (e.g. an intrusion detection system). We consider that these assumptions are not very far from reality since this is usually the setting under which an administrator/ISP needs to take a decision. He/she usually has access to a monitoring system and an intrusion detection system that has a detection rate and a false alarm rate. The actual intrusion detection/monitoring system is beyond the scope of this paper (e.g. a system as this proposed in Liu et al., 2014) and we consider that such a system is available to the administrator/ISP.

However, we consider that the ISP does not know the actual type of each node neither how much he gains or losses at every time step. Our theoretical and experimental analyses show that even this case (sub-class) of the problem is not that simple. We provide a novel algorithm – called High Probability Efficient Response algorithm (HiPER) – that could be employed for this efficient decision making and compare its performance with some approximations of MDP. We show that the performance of our algorithm is near optimal, even though the prior information required by HiPER is much less than actual MDP approximations.

The paper is organised as follows. In the remainder of this section we give some background and present related work. Section 3 introduces notation while Section 4 specifies the loss model. Section 5 presents the proposed HiPER algorithm as well as the bounds on the *worst-case expected loss*. Section 6 describes the decision-theoretic approaches which model the problem as an MDP and are used in the performance comparisons with the HiPER algorithm while Section 7 describes the evaluation experiments. Finally, Section 8 concludes the paper. The appendix provides proofs of technical lemmas and some useful auxiliary results.

2. Related work

As previously mentioned, our setting corresponds to a type of stopping problem. This has been extensively studied in general (DeGroot, 1970), while partial monitoring games in general have also received a lot of attention recently (Cesa-Bianchi and Lugosi, 2006). However, to the best of our knowledge, the *general hidden reward stopping problem* has not been previously studied in the literature. On the other hand, the specific application we consider can be seen as a type of *optimal intrusion response*.

The problem of intrusion response has received a lot of attention in the literature (Mitrokotsa et al., 2007a, 2007b). Most of the previous research on intrusion response has concentrated on the partially observable Markov decision processes (POMDP) formalism. Indicative publications are those by Zonouz et al. (2009), Zan et al. (2010), and Zhang et al. (2009), which have all proposed an intrusion response through modelling the process as a partially observable Markov decision process (POMDP) (Smallwood and Sondik, 1973). More precisely, Zonouz et al. (2009) proposed a Response and Recovery Engine (RRE)

based on a game-theoretic response strategy against adversaries modelled as opponents in a non-zero-sum, two-player Stackelberg stochastic game. In each step of the game RRE chooses the response actions using an approximate POMDP solver. More precisely, using the most likely state (MLS) (Cassandra, 1998) approximation, the POMDP is converted to a competitive Markov Decision Process (MDP), which is then solved using a look ahead search (i.e. approximate planning). Zhang et al. use the POMDP to integrate low level IDS alerts with high level system states, while Zan et al. (2010) propose to solve the intrusion response problem as a factored POMDP model. Additionally, they decompose the POMDP into small sub-POMDPs and compute the response policy using the MLS approximation technique. However, in our case MLS as an approximation is too crude to be used, since it would essentially result in a completely random policy, as there are only two possible hidden states each node can be in. An entirely different approach, policy-gradient methods, is employed by Dejmal et al. (2008) in the context of combating denial-of-service attacks in P2P networks. However, this approach requires observing the rewards, which are in fact hidden in our case.

2.1. *Our contributions*

Our contributions are three-fold. We provide four algorithms that could be employed to solve the blacklisting problem in an efficient way. Our first proposed algorithm called HIPER relies on bounds which do not require knowledge of prior probabilities regarding the type of a node (honest or malicious) neither known distributions for the observations corresponding to honest or malicious nodes. We only need to know the mean of each of these distributions. Consequently, it is substantially more lightweight than MDP solvers, since we take decisions without performing explicit planning. Thus, it is more suitable for resource constrained environments (e.g. wireless communications). We analyse the *expected loss* of this algorithm, and show that it is not significantly worse to that of an oracle which already knows each node's type.

Our *second contribution* is the proposal of three Bayesian decision-theoretic approaches that we derive by formalising the problem as a Markov decision process (MDP). In contrast to previous work, in our scenario the reward (actual gain or loss for the ISP provider) is *never observed* by the algorithm.[1] This corresponds to reality, since we frequently do not know which nodes give us negative rewards.[2] Furthermore, two of our MDP algorithms are different from those previously employed in the intrusion response literature, as we forego the most-likely-state approximation commonly used in POMDP approaches. We first consider a myopic approximation.[3] The second approach is a lightweight *optimistic* approximation that performs no planning, which is derived from upper bounds (Dimitrakakis, 2009) on Bayesian decision making in unknown MDPs (Duff, 2002). To our knowledge, this approach has not been used in similar problems before. Finally, we consider online planning with finite lookahead (DeGroot, 1970; Ross et al., 2008). This approach takes decisions which consider the impact of all our possible future actions up to some horizon. This approach has been employed in other applications such as dialogue modelling (Bui et al., 2006), autonomous underwater vehicle mapping (Saigol and University of Birmingham. School of Computer Science, 2009), preference elicitation (Boutilier, 1999, 2002) and sensor scheduling (He and Chong, 2004) in wireless sensor networks.

Finally, we compare the four proposed algorithms and evaluate them in different conditions. The results demonstrate that of these algorithms, an *optimistic approximation* has similar performance to that of HIPER, while a *finite lookahead approximation* has increased performance, at the cost of additional computation.

3. Preliminaries

We consider a set of nodes, which can be either honest or malicious. We assume there is a reliable way to obtain statistics for each node, such as an IDS that gives us a numerical score for each node. We denote by $\mathcal{Q}$ the set of all malicious nodes and by $\mathcal{U}$ the set of all honest nodes. We consider that there is an entity $\mathcal{E}$ (for instance an Internet service provider (ISP) or a network administrator) who gains some reward g_u (subscription of all registered clients/nodes) for each moment that an honest node remains in the network and has a cost ℓ_q (cost caused when a malicious node/client disrupts other clients or threats the ISP and its reputation) for each moment that a malicious node stays in the network. A node may be removed by $\mathcal{E}$ at any time, for example through blacklisting. However, re-inserting a removed node is not normally possible. Thus, we perform a worst case analysis since falsely removed nodes cannot be re-inserted.

We use N to denote the (possibly random) time at which $\mathcal{E}$ removes a node from the network. In addition, any honest node may leave the network at some (random) time H. Specifically, we assume that an honest node (subscriber of the ISP) may decide to leave the network with some small probability $\lambda > 0$, independently over time. Then it holds that $\mathbb{E}[H] = 1/\lambda$. We consider that the nodes are consistent with their behaviour which means that each node may be either honest or malicious. The type is hidden from $\mathcal{E}$; since the $\mathcal{E}$ (ISP or administrator) does not know if one of the clients/nodes is behaving honestly. This does not mean that a node cannot *behave* maliciously during one period and honestly the next: a node that behaves maliciously (e.g. drops packets on purpose) might not do so all the time. Of course, $\mathcal{E}$ does not only know the type of each node, but it also never observes the rewards obtained or the cost incurred (i.e. the actual loss due the possible threats and disruption of the network).

At each time-step t and for each node i, $\mathcal{E}$ receives an *information* signal $x_{i,t} \in [0, 1]$, characterising the behaviour of that node i within the time interval $t \in \mathbb{N}$. This signal can be seen as the output from some IDS, summarising the behaviour of that node during that period.

Assumption 1. *We assume that* $x_{i,1}, \ldots, x_{i,t}$ *are independent, (but not identically) distributed, random variables and:*

[1] Although of course the reward is used in the experiments to measure performance.

[2] Conversely, if we could observe the rewards, it would be trivial to identify malicious nodes.

[3] This is equivalent to the most likely state approximation and to a sequential probability ratio test under some conditions.

$$\mathbb{E}[x_{i,t}|\mathcal{Q}] = q, \quad \mathbb{E}[x_{i,t}|\mathcal{U}] = u. \tag{1}$$

While the expected value is constant for all t, the observed average of $\frac{1}{t}\sum_{k=1}^{t} x_{i,t}$ for each node i will initially be far from the expected value for small t. The average, together with the total number of observations for each node, forms a summary of the information received by each node. The relationship between these quantities will be looked at more closely in the analysis.

In an *intrusion response* scenario (e.g. Lee et al., 2000), q and u could be considered as the *detection rate* (DR) and the *false alarm rate* (FA) correspondingly of an employed intrusion detection system (e.g. Dimitrakakis and Mitrokotsa, 2009; Mitrokotsa and Karygiannis, 2008; Mitrokotsa et al., 2007c). Then $x_{i,t}$ would correspond to alarm signals, with lower and high values for innocent-looking and suspicious behaviour respectively. Correspondingly, in a peer-to-peer scenario (e.g. Si et al., 2009), they could be fairness or reputation scores of each node.

In the remainder, we always refer to some arbitrary node in the network and thus make no distinction between nodes. This is because the algorithms that we examine consider each node *independently* of the others. Consequently, the following section analyses the expected loss for a single node of unknown type.

4. The loss model

As previously mentioned, $\mathcal{E}$ obtains a small gain for each time-step an honest node is within the network, and a small loss for each time-step a malicious node remains in the network. Formally, we can write that the total gain G we obtain from some node i, which $\mathcal{E}$ removes at time N and which would voluntarily leave at time H is:

$$G(i, H, N) = \begin{cases} -N\ell_{\mathcal{Q}}, & i \in \mathcal{Q} \\ \min\{H, N\} g_{\mathcal{U}}, & i \in \mathcal{U}. \end{cases} \tag{2}$$

The above equation states that the gain depends on if the node is honest or malicious. If it is malicious then it depends on the loss we have from the malicious behavior $\ell_{\mathcal{Q}}$ of the node while if it is honest it depends on the subscription of the node $g_{\mathcal{U}}$ in the network and obviously in both cases there is a dependence on how long the node stays in the network.

$\mathcal{E}$ wants to choose some node removal policy π that maximises his total expected gain. That means that $\mathcal{E}$ needs to keep as many as possible honest nodes in the network and eliminate the nodes that behave maliciously. In our analysis, we compare the expected gain of our policy π with that of an *oracle*. The oracle always knows the type of each node (i.e. honest or malicious), and thus, employs the optimal policy π^*. For $i \in \mathcal{Q}$, according to the optimal policy π^* it holds $N = 0$, while for $i \in \mathcal{U}$ according to the optimal policy π^* it is $N = \infty$. Correspondingly,

$$\mathbb{E}_{\pi^*}[G(i)] = \begin{cases} 0, & i \in \mathcal{Q} \\ \mathbb{E}[H] g_{\mathcal{U}}, & i \in \mathcal{U}. \end{cases} \tag{3}$$

Let the loss L be the difference between the gain of the optimal policy and our policy. In particular, the *expected loss* of policy π for a node of type v is defined as:

$$\mathbb{E}_{\pi}[L|v] = \mathbb{E}_{\pi^*(v)}[G|v] - \mathbb{E}_{\pi}[G|v], \tag{4}$$

where the i subscript has been dropped for simplicity. The expected loss is bounded by the *worst-case expected loss*:

$$\mathbb{E}_{\pi}[L] \leq \max_{v \in \{\mathcal{Q}, \mathcal{U}\}} \mathbb{E}_{\pi}[L|v], \tag{5}$$

which we wish to minimise. If $\mathcal{E}$ removes node i from the network at random time N, then he does not receive any more observations $x_{i,t}$ for this node from the IDS. Thus, in essence, we want to find a *stopping rule*, that will let $\mathcal{E}$ to determine the random time N at which stopping occurs, i.e. $\mathcal{E}$ takes the decision that $i \in \mathcal{Q}$ and removes it from the network. We note that $0 \leq N \leq \infty$, where $N = \infty$ if stopping never occurs.

Since $\mathcal{E}$ does not know if node i is honest or malicious, it must collect a sufficient number of samples so as to only remove nodes for which it is reasonably certain that they are malicious. On the other hand, malicious nodes must be removed as soon as possible, since the operator (e.g. administrator/ISP) incurs a cost for every moment they remain in the network. The first algorithm we consider uses simple statistics to make nearly optimal decisions about which nodes to keep.

5. The HIPER algorithm

The algorithm, depicted in Alg. 1, uses the knowledge we have about malicious and honest nodes (see equation 1). This is done by calculating the average of all the observations (e.g. alarms received by the IDS) generated by a node i until time t:

$$\theta_t \triangleq \frac{1}{t}\sum_{k=1}^{t} x_{i,k}, \tag{6}$$

and adding an appropriate *confidence interval* so that errors are made with low probability. Informally, HIPER keeps nodes in the network as long as the statistic θ_t (e.g. average of observations/measurements provided by the IDS) is sufficiently far from the expected statistic q (e.g. detection rate of the IDS) of malicious nodes. In order to avoid throwing away honest nodes prematurely, it always keeps nodes for a certain number of steps to obtain more reliable statistics (e.g. observations/measurements provided by the IDS). However, as time passes, it needs more and more evidence to kick a node out. Consequently, the probability that an honest node is thrown out is bounded.

The analysis of the algorithm proceeds in three steps. First, we calculate the expected loss of the algorithm when faced with a node of malicious type. Then, we calculate the loss for honest nodes. Subsequently, we combine the two losses and tune the algorithm's input parameters to obtain an overall loss bound.

Algorithm 1 HiPER Algorithm for Optimal Response

Parameters: $\delta, \Delta, q \in [0,1]$
Loop: **For** each node i in the network:
For each time-step t do:
if $|\theta_t - q| < \sqrt{\frac{\ln(2/\delta)}{2t}}$ and $t > \frac{\ln(2/\delta)}{2\Delta^2}$ **then**
remove node i from the network
else keep node i in the network.
end if
end For
end For

The first bound only depends upon the input parameter δ, the error probability we wish to accept (it represents the tolerance of malicious behavior in the network), and the loss $\ell_{\mathcal{Q}}$ incurred by malicious nodes (e.g. loss because of disruption of the network, loss because of bad reputation of the ISP). We prove that the expected loss is polynomially bounded in terms of both δ and $\ell_{\mathcal{Q}}$.

Lemma 1. *For Algorithm 1, with input parameter δ, and $\Delta = |u - q|$, the* expected loss *when the node is malicious is bounded as:*

$$\mathbb{E}[L|\mathcal{Q}] \leq \frac{\ell_{\mathcal{Q}}}{(1-\delta)^2} \tag{7}$$

The proof of this lemma can be found in the appendix. Naturally, the expected loss is linearly dependent on the loss of keeping a malicious node in the network, while the dependence on the error probability is quadratic.

The second bound depends on the input parameter Δ, which corresponds to how far we expect the statistics of honest nodes to be from q, the gain obtained by honest nodes $g_{\mathcal{U}}$ and the leaving probability of honest nodes λ. Once more, we obtain a polynomial loss bound in terms of those variables.

Lemma 2. *If $\Delta = |u - q|$, then the* expected loss *when the node is honest is bounded by:*

$$\mathbb{E}[L|\mathcal{U}] \leq \frac{g_{\mathcal{U}}\delta}{2\lambda\left[1 - \exp\left(4\Delta - 2\Delta^2\right)\right]}. \tag{8}$$

The proof of this lemma can be found in the appendix. Similarly to the previous lemma, there is a linear dependence on the loss that is incurred when we erroneously remove an honest node, and a quadratic dependence on the rate of departure. In addition, there is a weak dependence on the gap Δ between the two means.

Finally, we can combine everything in one bound by selecting a value for δ that depends on Δ and which simultaneously makes the bounds tight:

Theorem 1. *Set $\Delta = |u - q|$ and select:*

$$\delta = \sqrt{\frac{2\ell_{\mathcal{Q}}}{g_{\mathcal{U}}}}\lambda^{1/3}. \tag{9}$$

If $\delta \leq \frac{1}{2}$, then the expected loss $\mathbb{E}L$ is bounded by:

$$\mathbb{E}(L) \leq \max\left\{\sqrt{\frac{\ell_{\mathcal{Q}} g_{\mathcal{U}}}{2}} f(\Delta), \frac{g_{\mathcal{U}}}{2}\right\}\lambda^{-2/3}, \tag{10}$$

where

$$f(\Delta) = \frac{1}{1 - \exp\left(4\Delta - 2\Delta^2\right)}.$$

is the difficulty in distinguishing the attacker.

Proof. For the first term, note that since

$$\delta = \sqrt{\frac{2\ell_{\mathcal{Q}}}{g_{\mathcal{U}}}}\lambda^{1/3} \leq 1/2,$$

then

$$\delta \leq 1 - \sqrt{\frac{2\ell_{\mathcal{Q}}}{g_{\mathcal{U}}}}\lambda^{1/3}.$$

Consequently, from Lemma 1 we have

$$\mathbb{E}[L|\mathcal{Q}] \leq \frac{\ell_{\mathcal{Q}}}{(1-\delta)^2} = \frac{g_{\mathcal{U}}}{2}\lambda^{-2/3}.$$

The second term follows directly from Lemma 2 by plugging in the selected value of δ:

$$\mathbb{E}[L|\mathcal{U}] \leq \frac{g_{\mathcal{U}}\delta}{2\lambda} f(\Delta) = \frac{\sqrt{2\ell_{\mathcal{Q}} g_{\mathcal{U}}}}{2} f(\Delta)\lambda^{-2/3}.$$

This theorem shows that the performance of HiPER only very weakly depends on the gap Δ between honest and malicious nodes. In addition, it is optimal up to a polynomial factor.

6. Markov decision process approximations

As mentioned in Section 1, our setting corresponds to an optimal stopping problem. As this can be modelled as MDP (DeGroot, 1970), it may be useful to solve the problem directly using the MDP formalism.

To cast our problem in this setting, we need to specify: (a) the *prior* probability for each node being *honest* or *malicious*; (b) a *known* distribution family for the observation distribution, conditioned on whether the node under consideration is honest or malicious; (c) a planning algorithm for calculating our responses. This can be quite demanding computationally, as the full solution to the problem requires planning in a large tree (Dimitrakakis, 2009; Kearns et al., 1999; Wang et al., 2005). However, they can result in much better performance.

6.1. *Intrusion response and POMDP*

A Partially Observable Markov Decision Process (POMDP) (Smallwood and Sondik, 1973) is a generalisation of a Markov Decision Process (MDP). More precisely, a POMDP models the relationship between an agent and its environment when the agent cannot directly observe the underlying state. A POMDP can be described as a tuple $\langle S, A, O, T, \Omega, R\rangle$ where S is a finite set of states, A is a set of possible actions, O is a set of possible

observations, T is a set of conditional transition probabilities and Ω is a set of conditional observation probabilities and $R: A, S \to \mathbb{R}$. .

We can model our intrusion response problem as a POMDP if we consider that a node of the network at each time-step t has a state $s_t \in S$ with $s_t = (v_t, c_t)$ where $v_t \in \{0, 1\}$ and $c_t \in \{0, 1\}$ such that:

$$v_t = \begin{cases} 0, & \text{if the node is honest,} \\ 1, & \text{if the node is malicious.} \end{cases}$$

$$c_t = \begin{cases} 0, & \text{if the node is in the network,} \\ 1, & \text{if the node is out of the network.} \end{cases}$$

where it holds that $\mathbb{P}(v_{t+1} = v_t) = 1$ since v_t is stationary.

Additionally, at each time-step t, $\mathcal{E}$ can perform an action $a_t \in \{0, 1\}$ such that:

$$a_t = \begin{cases} 0, & \text{if } \mathcal{E} \text{ keeps the node in the network,} \\ 1, & \text{if } \mathcal{E} \text{ removes the node from the network.} \end{cases}$$

Furthermore, the following independence condition holds: $\mathbb{P}(v_{t+1}|v_t, c_t, a_t) = \mathbb{P}(v_{t+1}|v_t)$ since the type of a node (i.e. malicious or honest) does not depend on $\mathcal{E}$'s action (i.e. remove from the network or not) neither on whether the node is in the network or out. In addition, since the type of a node never changes, it holds:

$$\mathbb{P}(v_{t+1} = j|v_t = j) = 1. \tag{11}$$

Consequently, we remove the time subscript from v in the sequel. On the other hand, the probability that a node will be in the network depends on if it is already in or out and the action that $\mathcal{E}$ will take:

$$\mathbb{P}(c_{t+1}|c_t, v, a_t) = \mathbb{P}(c_{t+1}|c_t, a_t) \tag{12}$$

From equations (11) and (12), it is evident that the POMDP under consideration is factored.

To fully specify the model we must assume some probability distribution for the observations. Specifically, we model x_t as drawn from a Bernoulli distribution[4] with parameters u and q for honest and malicious nodes respectively: $\mathbb{P}(x_t = 1|v = 0) = u$ and $\mathbb{P}(x_t = 1|v = 1) = q$. Let $\mathbf{x}_t \triangleq (x_1, \ldots, x_t)$ be a t-length sequence of observations. From Bayes' theorem, we obtain an expression for our *belief* at time t:

$$\mathbb{P}(v = j|\mathbf{x}_t) = \frac{\mathbb{P}(\mathbf{x}_t|v = j)\mathbb{P}(v = j)}{\sum_{i=0}^{1} \mathbb{P}(\mathbf{x}_t|v = i)\mathbb{P}(v = i)} \tag{13}$$

where $j \in \{0, 1\}$. Thus, the expected gain at time t if $\mathcal{E}$ decides to keep a node in the network is: $\mathbb{E}[G_t|c_t = 0, \mathbf{x}_t] = \mathbb{P}(v = 0|\mathbf{x}_t) \cdot g_u - \mathbb{P}(v = 1|\mathbf{x}_t) \cdot \ell_Q$ while the expected gain if $\mathcal{E}$ decides to remove the node from the network is always: $\mathbb{E}[G_t|c_t = 1] = 0$. The problem is to find a policy $\pi : X^* \to A$, mapping from the set of all possible sequences of observations to actions, maximising the total expected gain:

$$\mathbb{E}_\pi(G) = \mathbb{E}_\pi\left(\sum_{t=1}^{\infty} G_t\right). \tag{14}$$

Since future gains depend on any future observations we might obtain, the exact calculation requires enumerating all possible future observations. Consequently, the exact solution to the problem is intractable (DeGroot, 1970; Dimitrakakis, 2009; Duff, 2002). In the next section, we describe possible approximations to this problem.

6.2. POMDP algorithms

We consider three algorithms: (a) a *myopic* algorithm, which only considers the expected gain at the current time-step; (b) an *optimistic* algorithm, which computes an upper bound on the total expected gain; (c) a *finite lookahead* algorithm, which performs complete planning up to some fixed finite depth. While these algorithms have appeared before in the general MDP literature, they have not been applied before to intrusion response and blacklisting problems. We do not consider the most likely state approximation (MLS), since in our case there are only two possible hidden states for a node, thus, rendering the approximation far too coarse for it to be effective.

6.2.1. Myopic

In this case, $\mathcal{E}$ only considers the expected gain for the next time-step when taking a decision. Consequently, $\mathcal{E}$ keeps the node in the network if:

$$\mathbb{E}[G_t|a_t = 0] > \mathbb{E}[G_t|a_t = 1]. \tag{15}$$

This algorithm is the closest to the MLS approximation that is usually employed in the intrusion response systems, among the ones considered. In fact, it is easy to see that it would be identical to MLS, as well as to a sequential probability ratio test, when $\ell_Q = g_u$. The algorithm is also a special case of the finite look ahead algorithm, described in Section 6.2.3.

6.2.2. Optimistic

This rule constructs an upper bound on the value of the decision to keep a node in the network, which is based on Proposition 1 in Dimitrakakis (2009). Informally, this is done by assuming that the true type of the node will be revealed at the next time-step. Then $\mathcal{E}$ keeps the node in the network if and only if:

$$\mathbb{P}(v_t = 0|\mathbf{x}_t) \cdot g_u/\lambda > \mathbb{P}(v_t = 1|\mathbf{x}_t) \cdot \ell_Q. \tag{16}$$

Intuitively, if the node is revealed to be malicious, then we can remove it at the next step and consequently we only lose ℓ_Q. In the converse case, we can keep it for an expected $1/\lambda$ steps. This algorithm is closely related to Thompson sampling (Thompson, 1933), which has been extensively studied recently (Agrawal and Goyal, 2012; Kaufmanna et al., 2012; Osband et al., 2013) for the case of undiscounted bandit

[4] This distribution is particularly convenient for computational reasons, because closed-form Bayesian inference can be performed via the Beta conjugate prior (DeGroot, 1970). However, in principle it can be replaced with any other distribution family, without affecting the overall formalism.

problems and Markov decision processes. However, the Thompson sampling analysis does not directly carry over due to its stochasticity.

6.2.3. *Finite lookahead*

The finite lookahead algorithm performs backwards induction (DeGroot, 1970) up to some finite depth T, at every time-step. More precisely, any sequence of observations $\mathbf{x}_t = (x_1, \dots, x_t)$ results in a posterior probability $\mathbb{P}(v_t|\mathbf{x}_t)$. Let $V_t \triangleq \sum_{k=t}^{\infty} G_k$ be the total gain starting from time-step t. Then, the expected gain under the optimal policy is determined recursively as follows:

$$\mathbb{E}(V_t|\mathbf{x}_t) = \max\{0, \mathbb{E}(G_t|\mathbf{x}_t, a_t = 0) + \mathbb{E}(V_{t+1}|\mathbf{x}_t)\}$$

$$\mathbb{E}(V_{t+1}|\mathbf{x}_t) = p_t \mathbb{E}(G_t|\mathbf{x}_t, x_{t+1} = 1) + (1 - p_t)\mathbb{E}(V_{t+1}|\mathbf{x}_t, x_{t+1} = 0)$$

where $p_t \triangleq \mathbb{P}(x_{t+1} = 1|\mathbf{x}_t) = \sum_{i=0}^{1} \mathbb{P}(x_{t+1} = 1|v = i)\mathbb{P}(v = i|\mathbf{x}_t)$ is the marginal posterior probability that $x_{t+1} = 1$. For more details on this backwards induction algorithm, the reader is urged to consult DeGroot (1970) and Duff (2002).

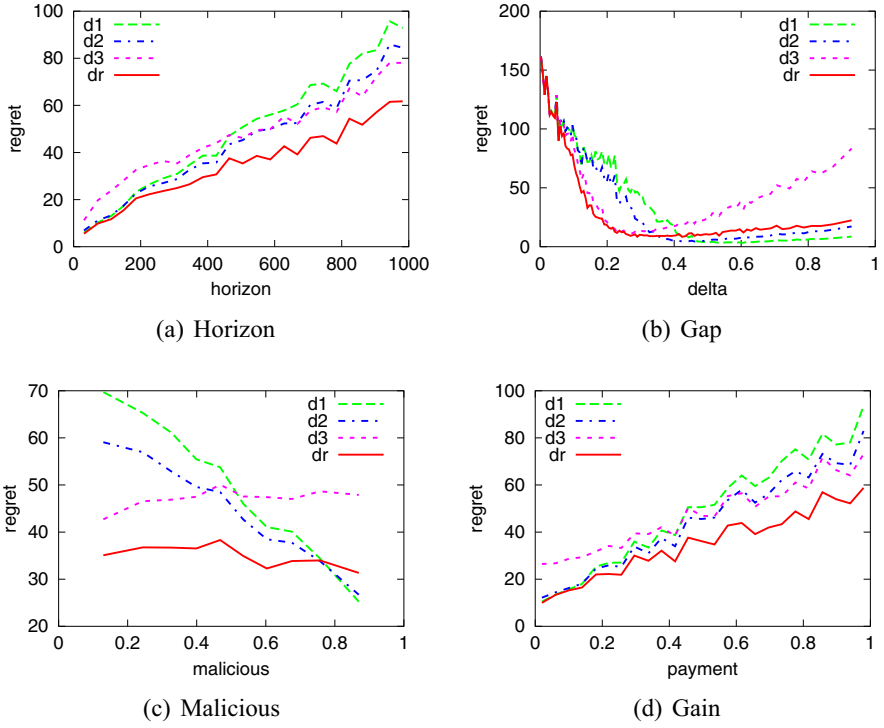
Since such algorithms do not consider the continuation, they are only optimal for the first T steps. In general, for a $(1-\lambda)$-discounted MDP (see for example Dimitrakakis, 2009) they achieve a performance that is $(1-\lambda)^T/(1-\lambda)$ close to the optimal. This is similar to our setting, where the probability λ of an honest node leaving the network is equivalent to a discount rate $(1-\lambda)$.

7. Experimental evaluation

We perform three sets of experiments. In all cases, we choose a range of possible problem parameters, drawn from some distribution. We then plot the expected performance of different algorithms, as one particular problem parameter changes, while averaging over the remaining problem parameters. For example, in Fig. 1(a) we plot the expected loss as the horizon increases, while other problem parameters, such as the gap of the proportion of malicious nodes and the gain, are random.

The first set of experiments investigates the performance of HIPER with various choices of the parameter δ. We compare this with the optimal choice suggested by Theorem 1 and show that this is indeed the best choice nearly always. We thus can choose the only parameter of the algorithm automatically. The second set compares HIPER with the *myopic* and *optimistic* approximations. In the final set of experiments, we compare the *optimistic* with the *finite lookahead* approximation. In all cases, we collected results from 10^4 runs, with 100 nodes in each simulation, and we plot a moving average of the *expected loss* as various network parameters change.

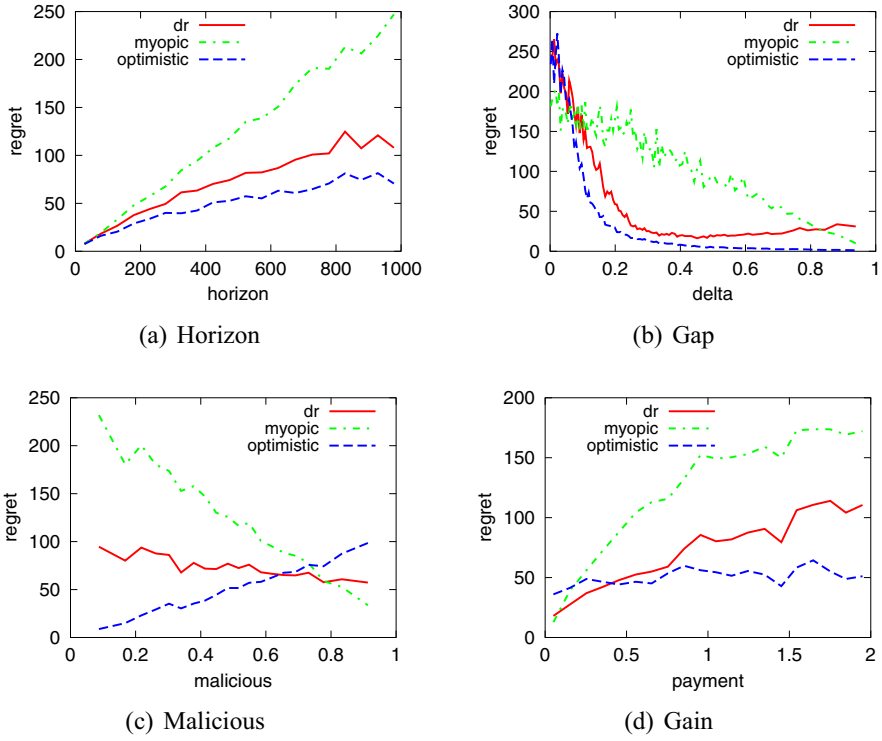
Specifically, the results we report (i.e. Fig. 1) are made through 10^4 experiments. For each experiment, we selected a horizon $H \sim Uniform([10, 1000])$, user and adversary parameters $u, q \sim Uniform([0, 1])$, and user gain $g_u \sim Uniform([0, 1])$ and we set $\ell_{\mathcal{Q}} = 1$. Each experiment measured the loss for a



(a) Horizon (b) Gap

(c) Malicious (d) Gain

Fig. 1 – Simulations with Alg. 1, for four different choices of δ. In particular $\delta_1 = 0.9$, $\delta_2 = 0.95$, $\delta_3 = 0.99$ and δ^* is chosen according to Theorem 1. It can be seen that, while the algorithm is not extremely sensitive to the exact choice of δ, the optimal value is generally more robust.



(a) Horizon (b) Gap

(c) Malicious (d) Gain

Fig. 2 – Comparison of HiPER with the *myopic* solver and the *optimistic* approximation for various network conditions. It can be clearly seen that the *myopic* approximation is significantly worse than both approaches. However, the *optimistic* approach outperforms the worst-case HiPER algorithm when the proportion of malicious nodes is low. The *optimistic* approach is also better when the payment for honest nodes is high.

network containing 100 nodes, each of which had a probability p of being malicious, with $p \sim Beta(2,2)$ for each experiment. During each run, the ith node generates a sequence of observations $x_{i,t}$ drawn from a Bernoulli distribution with parameter u if the node is honest and q if the node is malicious.

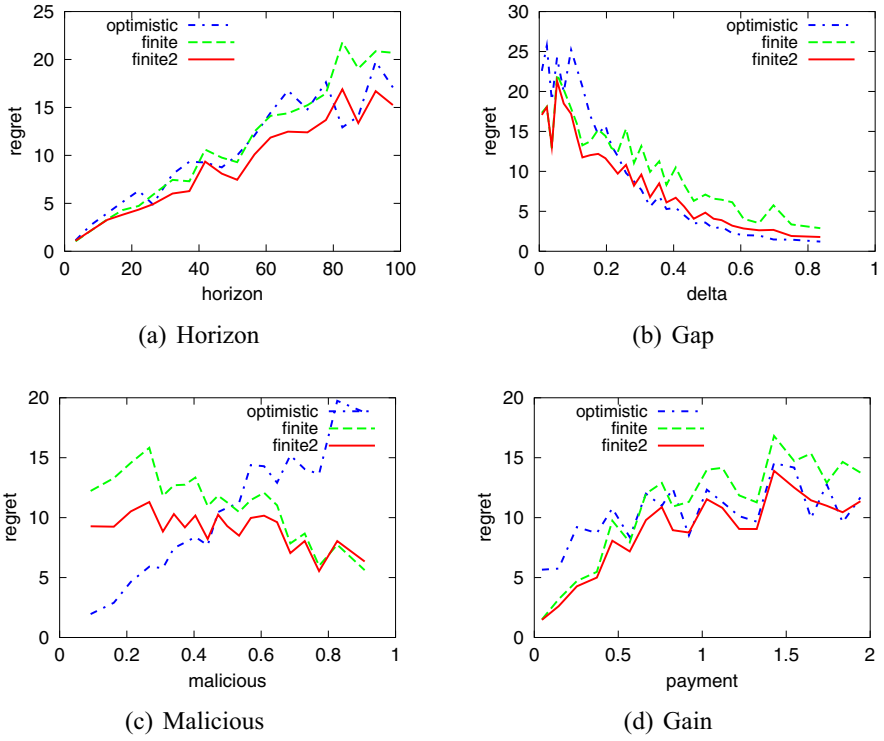
The first set of results examines the choice of δ in the theorem compared to alternatives for HiPER. More precisely, Fig. 1 shows a summary of the results, averaged over these trials. It can be seen that, while HiPER's performance is relatively robust to the choice of δ, nevertheless the optimal choice suggested by Theorem 1 generally leads to small losses.

For our second set of experiments, shown in Fig. 2, we compare HiPER with the *optimistic* and *myopic* algorithms. We increased the range of user gains to $g_u \sim Uniform([0,2])$ compared to the previous setup, but the other experimental parameters remain the same. It is clear that the *myopic* approximation has almost always a higher loss compared to both HiPER and the *optimistic* algorithm. The latter, while performing at a similar level to HiPER, has an advantage when either the proportion of malicious is small or when g_u is large. This makes sense intuitively, since in those cases the optimism is justified. In the converse case, however, the *optimistic* approach performs worse than HiPER, which is less sensitive to the proportion of malicious nodes, since it is a worst-case approach.

Finally, we performed some experiments comparing the *optimistic* approximation with the *finite-lookahead* POMDP solvers for lookahead for T time-steps where $T \in \{4, 8\}$. While these do not solve the problem to the end of the horizon H, they plan ahead for T steps at every time-step of the simulation. Unfortunately, the complexity of these solvers is exponential in T, which limited the amount of simulations we could perform to 10^3 and we only considered horizons $H \sim Uniform([1,100])$. These experiments are shown in Fig. 3. In comparison with Fig. 2, the *finite lookahead* algorithms performs much better than the *myopic* approximation and indeed the 8-step lookahead manages to slightly outperform the *optimistic* approximation. In addition, it is much more robust to the proportion of malicious nodes in the network. However, the relative advantage of the 8-step to the 4-step lookahead is relatively small for the amount of extra computation required.[5]

Overall, the myopic approximation, which a generalisation of the MLS approximation commonly used in intrusion response, performs significantly worse than any other algorithm. First of all, this is due to the fact that this is the 0-horizon case of the lookahead algorithm, so it has to be worse than any instantiation of that. Nevertheless, the optimistic approximation, which employs an upper bound, is nearly as good as the lookahead approximation. This is because the optimistic approximation prioritises the collection of information, which is important for this problem. This is something that is

[5] The computational effort is exponential in T.



(a) Horizon (b) Gap

(c) Malicious (d) Gain

Fig. 3 – Comparison of the optimistic approximation with approximate *non-myopic* POMDP solvers for planning lookahead of T time-steps where $T \in \{4, 8\}$. It can be seen that, for short horizons, these perform just as well and that they are more robust to the proportion of malicious nodes in the network. However, these methods are computationally more intensive, with complexity $O(e^T)$.

well-known from the bandit literature, where optimistic methods in bandit problems, such as upper confidence bounds algorithms (Auer et al., 2002), have near-optimal performance. The same holds for HiPER itself, which can also be seen as an upper confidence bound approach, that is specifically tuned for this type of problem, rather than bandit problems.

8. Conclusion

This paper addresses the decision making problem on blacklisting nodes for efficient network management; a problem that arises frequently in communication networks, namely, whether to remove a suspicious node from the system, with the amount of available evidence, or to collect some further data before taking the final decision. This is in fact a type of stopping problem, which we believe is of relevance to many applications where blacklisting may be performed. This includes applications such as automated intrusion response, as well as ensuring fairness in peer-to-peer networks, such as Vieira et al. (2009). To this end, we proposed and analysed, both theoretically and experimentally, an efficient algorithm, HiPER, that achieves low *worst-case expected loss* relative to an oracle that knows *a priori* the type (honest or malicious) of every node in the system. In addition, we derived and compared a number of algorithms by modelling the problem as a POMDP: a *myopic* and an *optimistic* approximation, as well as a *finite lookahead* solver. Of those, the *optimistic* approximation and the partial *finite lookahead* solvers perform the best, with the *finite lookahead* methods being the most robust, while simultaneously being computationally demanding.

The main advantage of HiPER are its simplicity and lack of stringent assumptions on the distribution. This makes it suitable for deployment in most situations. However, whenever a full probabilistic model and computational resources are available, one of the approximate solvers would be useful. The overall best performance is offered by the *finite lookahead*, closely followed by the *optimistic* approximation. The *myopic* approximation, which is equivalent to the widely-used "most likely state" (MLS) approximation, is the worst. To our knowledge, neither the *optimistic* approximation nor the *finite lookahead* methods have been applied before to this problem or more generally to intrusion response problems. They should be more generally applicable for other types of intrusion response and resource management problems. It is our view that they are inherently more suitable than other approximations such as the commonly used (MLS) approximation (or equivalently, a sequential probability ratio test) which in our setting produces an essentially random policy.

For future work, we would like to extend our theoretical analysis to the performance of the *optimistic* and the *finite lookahead* algorithms. The first algorithm has so far not been analysed for MDPs in general. The latter algorithm has so far been analysed in the bandit and MDP setting, but only in how optimal the decision at each step is, rather than the overall loss. In addition, it would be interesting to examine more general game-theoretic

scenarios, including strategic attackers (Bao et al., 2011; Dritsoula et al., 2012). Finally, we would like to generalise our setting so that observations must be *explicitly* gathered from each node, where it is not possible to continuously sample all nodes due to budget constraints. In fact, the sampling problem in the context of intrusion detection has been recently studied by Bu et al. (2011) and Liu and Zhao (2011). A natural extension of our work would consequently be to optimally combine sampling and response policies.

Appendix A. Proofs

This section collects the missing proofs from the main text.

Proof (Proof of Lemma 1). Since the node i under consideration is malicious, i.e. $i \in \mathcal{Q}$, it holds that: $\mathbb{E}[x_{i,t}|\mathcal{Q}] = q$. Then, we have:

$$\mathbb{E}[\theta_t|\mathcal{Q}] = \mathbb{E}\left[\frac{1}{t}\cdot\sum_{k=1}^{t} x_{i,k}|\mathcal{Q}\right] = \frac{1}{t}\sum_{k=1}^{t}\mathbb{E}[x_{i,k}|\mathcal{Q}] = \frac{1}{t}\cdot t\cdot q = q.$$

From Hoeffding's inequality (*Lemma* 3, in the Appendix), we have:

$$\mathbb{P}(|\theta_t - q| > \varepsilon_t|\mathcal{Q}) \le 2\exp(-2t\varepsilon_t^2), \tag{A.1}$$

where $\mathcal{E}_t > 0$ and $\mathbb{P}(|\theta_t - q| > \varepsilon_t|\mathcal{Q})$ denotes the probability that θ_t (which is random) is very far away from q (which is fixed). Now set:

$$\varepsilon_t = \sqrt{\frac{\ln(2/\delta)}{2t}}$$

as in Algorithm 1. Then, since equation A.1 holds for any $\mathcal{E}_t > 0$, we get that the probability of keeping a malicious node $i \in \mathcal{Q}$ in the network is at most δ:

$$\mathbb{P}\left(|\theta_t - q| > \sqrt{\frac{\ln(2/\delta)}{2t}}\middle|\mathcal{Q}\right) < \delta.$$

Thus, we have:

$$\begin{aligned}\mathbb{E}[L|\mathcal{Q}] &= \mathbb{E}[N|\mathcal{Q}]\cdot\ell_{\mathcal{Q}} = \sum_{t=0}^{\infty}\mathbb{P}(N=t|\mathcal{Q})\cdot t\cdot\ell_{\mathcal{Q}}\\ &\le \ell_{\mathcal{Q}}\sum_{t=0}^{\infty}\delta^{t-1}\cdot t = \frac{\ell_{\mathcal{Q}}}{(1-\delta)^2}\end{aligned}$$

Proof (Proof of Lemma 2). We denote by N the time-step at which $\mathcal{E}$ removes node i from the network. Then, the function $g: \mathbb{N}^2 \to \mathbb{R}$ that gives us the gain for each node i is defined as: $g(n,h) \triangleq \min\{n,h\}\cdot g_{\mathcal{U}}$ where $h \in H$ and $n \in N$. Since the node i under consideration is honest, i.e. $i \in \mathcal{U}$, we have $\mathbb{E}[x_{i,t}|\mathcal{U}] = u$. Without loss of generality we assume that: $u = q + \Delta$, where $\Delta > 0$. So we only need $\mathbb{P}(\theta_t - q < \varepsilon_t|\mathcal{U})$. Since $q = u - \Delta$ from the Hoeffding inequality (*Lemma* 3, in the Appendix), we have:

$$\begin{aligned}\mathbb{P}(N=t|\mathcal{U}) &\le \mathbb{P}(N\le t) \le \mathbb{P}(\theta_t - u < \varepsilon_t - \Delta|\mathcal{U})\\ &\le \exp\left(-2\cdot t(\varepsilon_t-\Delta)^2\right)\end{aligned}$$

where $\Delta - \varepsilon_t > 0$. It holds that:

$$\mathbb{E}[G|\mathcal{U}, N=n] = \sum_{h=0}^{\infty}\mathbb{P}(H=h|\mathcal{U}, N=n)\mathbb{E}[G|\mathcal{U}, N=n, H=h] \tag{A.2}$$

But it holds that: $\mathbb{E}[G|\mathcal{U}, N=n, H] = g(n,h)$ and since $h \in H$ and $n \in N$ are independent we have: $\mathbb{P}(H=h|\mathcal{U}, N=n) = \mathbb{P}(H=h|\mathcal{U})$. Thus,

$$\begin{aligned}\mathbb{E}[G|\mathcal{U}, N=n] &= \sum_{h=0}^{\infty}\mathbb{P}(H=h|\mathcal{U})\cdot g(n,h)\\ &= \sum_{h=0}^{\infty}\mathbb{P}(H=h|\mathcal{U})\min\{n,h\}\cdot g_{\mathcal{U}}\\ &= g_{\mathcal{U}}\cdot\left\{\sum_{h=0}^{n-1}\mathbb{P}(H=h|\mathcal{U})\cdot h + \sum_{h=n}^{\infty}\mathbb{P}(H=h|\mathcal{U})\cdot n\right\}\end{aligned} \tag{A.3}$$

The expected loss is given by subtracting from the expected gain of the oracle policy, when $\mathcal{E}$ never removes the node from the network (i.e. $N=\infty$), the expected gain when $\mathcal{E}$ removes the node at the time-step $N=n$. Thus, it holds:

$$\begin{aligned}\mathbb{E}[L|\mathcal{U}, N=n] &= \mathbb{E}[G|\mathcal{U}, N=\infty] - \mathbb{E}[G|\mathcal{U}, N=n]\\ &= \lim_{n\to\infty}(\mathbb{E}[G|\mathcal{U}, N=n]) - \mathbb{E}[G|\mathcal{U}, N=n]\\ &= g_{\mathcal{U}}\sum_{h=0}^{\infty}\mathbb{P}(H=h)h - g_{\mathcal{U}}\left\{\sum_{h=0}^{n-1}\mathbb{P}(H=h)h + \sum_{h=n}^{\infty}\mathbb{P}(H=h)n\right\}\\ &= g_{\mathcal{U}}\left\{\sum_{h=n}^{\infty}\mathbb{P}(H=h)h - \sum_{h=n}^{\infty}\mathbb{P}(H=h)n\right\}\end{aligned} \tag{A.4}$$

Since, by definition $\mathbb{P}(H=h+1|H>h) = \lambda$, we have $\mathbb{P}(H=h) = (1-\lambda)^{h-1}\lambda$. Consequently,

$$\begin{aligned}\mathbb{E}[L|\mathcal{U}, N=n] &= g_{\mathcal{U}}\lambda\left(\sum_{h=n}^{\infty}(1-\lambda)^{h-1}h - \sum_{h=n}^{\infty}(1-\lambda)^{h-1}n\right)\\ &= g_{\mathcal{U}}\frac{(1-\lambda)^n}{\lambda}\end{aligned} \tag{A.5}$$

Thus, we have:

$$\begin{aligned}\mathbb{E}[L|\mathcal{U}] &= \sum_{t=0}^{\infty}\mathbb{P}(N=t|\mathcal{U})\mathbb{E}[L|N=t]\\ &\le \sum_{t=0}^{\infty}\exp\left(-2\cdot t\cdot(\varepsilon_t-\Delta)^2\right)\cdot g_{\mathcal{U}}\frac{(1-\lambda)^t}{\lambda}\end{aligned}$$

Since the algorithm uses

$$\varepsilon_t = \sqrt{\frac{\ln(2/\delta)}{2t}},$$

we have:

$$\begin{aligned}E[L|\mathcal{U}] &\le \frac{g_{\mathcal{U}}}{\lambda}\sum_{t=0}^{\infty}\exp\left(-2\cdot t\left[\sqrt{\frac{\ln(2/\delta)}{2t}} - \Delta\right]^2\right)(1-\lambda)^t\\ &= \frac{g_{\mathcal{U}}}{\lambda}\sum_{t=0}^{\infty}\exp(-\ln(2/\delta) + 4\Delta t - 2t\Delta^2)(1-\lambda)^t\\ &= \frac{g_{\mathcal{U}}\delta}{2\lambda}\sum_{t=0}^{\infty}\exp((4\Delta - 2\Delta^2)t)(1-\lambda)^t\\ &= \frac{g_{\mathcal{U}}\delta}{2\lambda}\frac{1}{1-1-\lambda)\exp(4\Delta-2\Delta^2)}\end{aligned} \tag{A.6}$$

$$\le \frac{g_{\mathcal{U}}\delta}{2\lambda}\frac{1}{1-\exp(4\Delta-2\Delta^2)}$$

where $t \ge 2$.

Appendix B. Additional results

Definition 1. (Bernoulli distribution). If $X_1, \ldots, X_n$ are independent Bernoulli random variables with $X_k \in \{0, 1\}$ and $\mathbb{P}(X_k = 1) = \mu$ for all k, then

$$\mathbb{P}\left(\sum_{k=1}^{n} X_k \geq u\right) = \sum_{k=0}^{u} \binom{n}{k} \mu^k (1-\mu)^{n-k}. \tag{B.1}$$

Lemma 3. (Hoeffding). For independent random variables $X_1, \ldots, X_n$ such that $X_i \in [a_i, b_i]$, with $\mu_i \triangleq \mathbb{E}X_i$ and $t > 0$:

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i \geq \sum_{i=1}^{n} \mu_i + nt\right) \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

The same in equality holds for $\sum_{i=1}^{n} X_i \leq \sum_{i=1}^{n} \mu_i - nt$.

REFERENCES

Agrawal S, Goyal N. Analysis of Thompson sampling for the multiarmed bandit problem. In COLT 2012, 2012.

Auer P, Cesa-Bianchi N, Fischer P. Finite time analysis of the multiarmed bandit problem. Mach Learn 2002;47(2/3):235–56.

Bao N, Kreidl P, Musacchio J. A network security classification game. In GameNets 2011, 2011.

Boutilier C. A POMDP formulation of preference elicitation problems. In Proceedings of the national conference on artificial intelligence, pages 239–46. Menlo Park, CA; Cambridge, MA; London: AAAI Press; MIT Press; 1999, 2002.

Bu S, Yu F, Liu X, Tang H. Structural results for combined continuous user authentication and intrusion detection in high security mobile ad-hoc networks. IEEE Trans Wireless Commun 2011;99:1–10.

Bui T, Poel M, Nijholt A, Zwiers J. A tractable DDN-POMDP approach to affective dialogue modeling for general probabilistic frame-based dialogue systems. 2006.

Cassandra A. Exact and approximate algorithms for partially observable Markov decision processes. [Ph.D. thesis]. Brown University; 1998.

Cesa-Bianchi N, Lugosi G. Prediction, learning and games. 2006.

DeGroot M. Optimal statistical decisions. John Wiley & Sons; 1970 Republished in 2004.

Dejmal S, Fern A, Nguyen T. Reinforcement learning for vulnerability assessment in peer-to-peer networks. In Proceedings of the 20th national conference on innovative applications of artificial intelligence, p. 1655–62, 2008.

Dimitrakakis C. Complexity of stochastic branch and bound methods for belief tree search in Bayesian reinforcement learning. In 2nd international conference on agents and artificial intelligence (ICAART 2010), p. 259–64, Valencia, Spain; 2009. ISNTICC, Springer.

Dimitrakakis C, Mitrokotsa A. Statistical Decision Making for Authentication and Intrusion Detection. In: Proceedings of the 8th IEEE International Conference on Machine Learning and Applications (ICMLA 2009). Miami, FL, USA: IEEE Computer Society; 2009. p. 409–14.

Dritsoula L, Loiseau P, Musacchio J. A game-theoretical approach for finding optimal strategies in an intruder classification game. In CDC 2012, 2012.

Duff MO. Optimal learning computational procedures for Bayes-adaptive Markov decision processes [Ph.D. thesis]. University of Massachusetts at Amherst; 2002.

He Y, Chong K. Sensor scheduling for target tracking in sensor networks. In 43rd IEEE Conference on decision and control, 2004. CDC, vol. 1, p. 743–8. IEEE; 2004.

Kaufmanna E, Korda N, Munos R. Thompson sampling: An optimal finite time analysis. In ALT-2012, 2012.

Kearns MJ, Mansour Y, Ng AY. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In Journal of Machine Learning 2002;49(23):193–208.

Lee W, Fan W, Millerand M, Stolfo S, Zadok E. Toward cost-sensitive modeling for intrusion detection and response. J Comput Secur 2000;10:5–22.

Liu K, Zhao Q. Dynamic intrusion detection in resource-constrained cyber networks. Technical Report arXiv:112.0101, 2011.

Liu L, Saha S, Torres R, Xu J, Tan P-N, Nucci A, et al. Detecting malicious clients in isp networks using http connectivity graph and flow information. In International Conference on advances in social networks analysis and mining (ASONAM), 2014 IEEE/ACM, p. 150–7, Aug 2014.

Mitrokotsa A, Dimitrakakis C, Douligeris C. Intrusion detection using cost-sensitive classification. In: Proceedings of the 3rd European conference on computer network defense (EC2ND 2007). Lecture notes in electrical engineering. Heraklion, Crete, Greece: Springer-Verlag; 2007c. p. 35–46.

Mitrokotsa A, Karygiannis A. Chapter: intrusion detection techniques in sensor networks. In: Book: wireless sensor network security. Cryptology and information security series. IOS Press; 2008. p. 251–72.

Mitrokotsa A, Komninos N, Douligeris C. Intrusion detection and response in ad hoc networks. International journal on computer research, special issue on advances in ad hoc network security, Nova Science Publishing Inc. 2007a; 15(1):23–55.

Mitrokotsa A, Komninos N, Douligeris C. Towards an effective intrusion response engine combined with intrusion detection in ad hoc networks. In: Proceedings of the 6th annual mediterranean ad hoc networking workshop (Med-Hoc-Net 2007). Corfu, Greece: 2007b. p. 137–44.

Osband I, Russo D, Roy BV. (More) efficient reinforcement learning via posterior sampling. In: NIPS. 2013.

Ross S, Pineau J, Paquet S, Chaib-draa B. Online planning algorithms for POMDPs. J Artif Intell Res 2008;32:663–704.

Saigol Z, University of Birmingham. School of Computer Science. Information-lookahead planning for AUV mapping. School of Computer Science, University of Birmingham; 2009.

Si P, Yu F, Ji H, Leung V. Distributed sender scheduling for multimedia transmission in wireless mobile peer-to-peer networks. IEEE Trans Wireless Commun 2009;8(9):4594–603.

Smallwood R, Sondik E. The optimal control of partially observable Markov processes over a finite horizon. Oper Res 1973;21:1071–88.

Thompson W. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika 1933;25(3–4):285–94.

Vieira A, Campos S, Almeida J. Fighting attacks in P2P live streaming: Simpler is better. In: INFOCOM Workshops 2009, IEEE. IEEE; 2009. p. 1–2.

Wang T, Lizotte D, Bowling M, Schuurmans D. Bayesian sparse sampling for on-line reward optimization. In Proceedings of the 22nd international conference on machine learning, p. 956–63. ACM; 2005.

Zan X, Gao F, Han J, Liu X, Zhou J. A hierarchical and factored pomdp based automated intrusion response framework. In Proceedings of the 2nd international conference on software

technology and engineering (ICSTE), vol. 2, p. 410–4. IEEE; 2010.
Zhang Z, Ho P-H, He L. Measuring IDS-estimated attack impacts for rational incident response: a decision theoretic approach. Comput Secur 2009;28:605–14.
Zonouz S, Khurana H, Sanders W, Yardley TM. RRE: A Game-Theoretic Intrusion Response and Recovery Engine. In Proceedings of the IEEE/IFIP International conference on dependable systems & networks, 2009 (DSN'09), p. 439–48, Lisbon, Portugal, 29 June–2 July, 2009.

Christos Dimitrakakis is currently a visitor associate professor at the Department of Mathematical and Computing Sciences at Tokyo Institute of Technology and affiliated with the Department of Computer Science and Engineering at Chalmers University of Technology. Formerly he was a Marie Curie Fellow at EPFL. His main research interest is decision theory, including reinforcement learning and problems in security applications. He obtained his PhD in 2006 from EPFL and has been a researcher at the University of Amsterdam and finally the Frankfurt Institute for Advanced Studies. He has previously co-organised workshops at CCS, ICML and ECML on privacy, security and machine learning.

Aikaterni Mitrokotsa is currently a visitor associate professor at Department of Mathematical and Computing Sciences at Tokyo Institute of Technology and also affiliated with the Department of Computer Science and Engineering at Chalmers University of Technology. Her main research interests lie in information security, privacy-preservation, machine learning for security and provable security. She has been awarded young researcher grant from the Swedish Research Council, the Rubicon Research Grant by NWO and a Marie Curie Intra European Fellowship. She has served on the PCs of INFOCOM, ACNS, Indocrypt, has co-organised a workshop at CCS, ECML on Privacy, Security, and Machine Learning.