Threshold FlipThem: When the winner does not need to take all

David Leslie[1], Chris Sherfield[2], and Nigel P. Smart[2]

[1] Department Mathematics and Statistics, University of Lancaster,
d.leslie@lancaster.ac.uk,
[2] Department of Computer Science, University of Bristol, UK,
c.sherfield@bristol.ac.uk, nigel@cs.bris.ac.uk.

Abstract. We examine a FlipIt game in which there are multiple resources which a monolithic attacker is trying to compromise. This extension to FlipIt was considered in a paper in GameSec 2014, and was there called FlipThem. Our analysis of such a situation is focused on the situation where the attacker's goal is to compromise a threshold of the resources. We use our game theoretic model to enable a defender to choose the correct configuration of resources (number of resources and the threshold) so as to ensure that it makes no sense for a rational adversary to try to attack the system. This selection is made on the basis of the relative costs of the attacker and the defender.

1 Introduction

At its heart security is a game played between an attacker and a defender; thus it is not surprising that there have been many works which look at computer security from the point of view of game theory [1, 9, 12, 15]. One particularly interesting example is the FlipIt game developed by van Dijk et al [16]. In FlipIt the attacker and defender are competing to control a resource. Both players are given just a single button each. The attacker gets control of the resource by pressing her button, whilst the defender can regain control by pressing his button. Pressing the button has a cost for each player, and owning the resource has a gain.

In this work we examine the FlipIt game in the situation where the defender has multiple resources, and the attacker is trying to obtain control of as many of these resources as possible. This was partially considered before in the paper [7], who introduced a variant of FlipIt called FlipThem in which the defender has control of multiple resources. Instead of flipping the state of a single resource from good to bad, the attacker is trying to flip the states of multiple resources. In [7] the authors examine the simplest situations in which an attacker "wins" if he has control of all resources, and a defender "wins" if she has control of at least one resource. Thus using the terminology of secret sharing schemes the paper [7] considers only the *full threshold* situation.

In this paper we study non-full threshold cases. This is motivated by a number of potential application scenarios which we now outline:

– Large web sites usually have multiple servers responding to user requests so as to maintain high availability and response times. An APT attack on a web site may

try to knock out a proportion of the servers so as to reduce the owners quality of service below an acceptable level.

- Large networks contain multiple paths between different nodes; again to protect against attacks. An attacker will not usually be successful if he knocks out a single path, however knocking out all paths is overkill. There will be a proportion of the paths which will result in a degradation of the network connectivity which the attacker may want to achieve.
- In many computer systems multiple credentials are needed to access a main resource. Thus an attacker only needs to obtain enough credentials to compromise a main resource. Thus modelling attacks on credentials (e.g. passwords, certificates, etc) should really examine the case of multiple credentials in the non-full threshold case.
- Multi-Party Computation (MPC) has always used threshold adversaries; an external attacker trying to compromise a system protected with MPC technology will only be interested in obtaining a threshold break above the tolerance limit of the MPC system. In such a situation however one is interested in proactively secure MPC systems, since when modelled by FlipThem a defender may regain control of a compromised party.
- Related to the last point is that of fault tolerance. It is well known that Byzantine agreement is not possible if more than $n/3$ of the parties are compromised. Thus an adversary who simply wants to inject errors into a network protected by some Byzantine agreement protocol only needs to compromise more than $n/3$ of the servers.

Thus we examine variants of the FlipThem game of [7] in which an attacker is trying to obtain control of at least t of the resources. We call this the (n, t)-FlipThem game.

Our main results are to examine Nash equilibria in the case of stochastic models of play. These are models in which the players strategies are defined by some random process. The random process defines, for each player, the next time point at which it will make a play (with time being considered as continuous). In all of the models we consider, we think of an attacker's play as being to attack a single resource; in the case of a stealthy defender the machine to attack at a given point in time will be random, whereas in the case of a non-stealthy defender the attacker will always attack a non-compromised resource. For the defender we allow two possible moves; in the first type the defender gains control of *all* resources with a single play. This models the situation where a defender might reset and reinstall a whole cloud infrastructure in one go, or reset all credentials/passwords in a given move; we call this a *full reset*. In the second type of move the defender needs to select a single resource to reset. Just like in the case of the attacker, the defender can do this in two ways depending on whether the attacker is stealthy or not. We call this type of defender move a *single reset*. This paper introduces continuous time Markov chains as a method of finding the benefit functions and calculating Nash equilibria of the two player partial threshold multi-party FlipIt game, FlipThem. For full reset, it finds that the equilibria depend solely on the threshold of the resources and the costs of play, not the number of resources involved. As the cost for the attacker increases the necessary amount of servers (threshold) required for the

defender to maximise his benefit decreases. For single reset, the analysis is harder by hand. However, using numerical methods, one can find analogous results.

1.1 Prior Work

The FlipIt game has attracted attention as it focuses on the situation where the attacker always gets in; building on the modern appreciation that perimeter defence on its own is no longer enough. For example the paper [2] examines the FlipIt game as applied to various different situations in computer security; for example password reset strategies, key management, cloud auditing and virtual machine refresh methodologies.

Despite its simplicity the FlipIt game is rather complex in terms of the possible different attacker and defender strategies, and can be modified in various ways. In the original FlipIt game both the attacker and the defender are 'stealthy' in the sense that neither knows if the other controls the resource before they execute a button press. In [13] the authors introduce a new mechanism where by a player can test who controls the resource. The idea being to model the situation whereby investigating whether a breach has occured is less costly than clearing up after a breach. Thus a 'peek'/'probe' at the resource state costs less than taking control of the resource. The paper [13] then moves onto discuss situations where a resource becomes hardened over time; meaning that every time a player moves on a resource he already controls, part of the move consists of making it harder for the opponent to regain control of the resource. An example would be a system administrator resetting the system to regain control and then patching the system so the attacker can not use the same method of infiltration.

One can think of the 'peek'/'probe' at the resource state from [13] as a way of removing the stealthiness from the FlipIt game. In [8] a different approach is proposed in which *defender* moves are not stealthy, i.e. an attacker knows if the defender controls the resource. This is introduced to model situations such as password resetting, in which an attacker knows when the password is reset (as he is no longer able to login), but the defender may not notice that their password is compromised. As well as this non-stealthy mode of operation the paper also introduces the idea of a defender trying to defend against multiple (independent) attackers.

The main prior work related to the current paper is that of Laszka et al [7]. They consider the same situation as us of multiple resources being attacked by a single monolithic adversary. However, their work has a number of distinct differences. Firstly, and most importantly, they focus on the case where an attacker wins if he controls all resources, and the defender wins when he controls one resource. We on the other hand examine a general threshold structure. Secondly, the paper of Laszka et al considers two types of strategies defined by periodic and non-arithmetic renewal processes[1]. The paper establishes some basis facts on these strategies, but does not consider constructing full benefit functions for either of these strategies and nor does it find analytic Nash equilibria for the strategies. This is due to the analytic difficulty in obtaining such formulae.

[1] A renewal process is called non-arithmetic if there is no positive real number $d > 0$ such that the inter-arrival times are all the integer multiples of d.

Given this (relatively) negative result the paper moves onto considering strategies arising from Markov processes. They develop a model for two resources, considering discrete time steps and set up a linear programming solution that becomes more complicated as the finite time horizon extends. We on the other hand are able to obtain simpler analytic formulae by considering a continous Markov process. This is because in [7] when constructing the Markov chain, they consider the state space to be the inter-arrival times of each resource with respect to the attacker.

In our paper we set up the state space to be the number of resources compromised at a specific (continuous) time. Thus moving from discrete to continuous time, and Markov to Stochastic processes simplifies the analysis somewhat. Without this simplification the paper [7] looks at two specific examples; trying to find the optimal strategy of the attacker given the strategy of the defender, and then the optimal flip rates that maximise the benefit at the defender side given that the attacker plays optimally. Finally they briefly mention how to find a Nash equilibrium, stating there is a simple iterative algorithm to find one but they state that algorithm will not converge for the majority of cases.

The paper [17] also considers a number of extensions of the FlipIt paper, and much like that of Laszka et al comments on the difficulty of obtaining analytic solutions to the Nash equilibrium. Therefore, they adopt a simulation based method. The attackers probability of compromising increases progressively with probing, while the defender uses a moving-target technique to erase attacker progress. The paper extends the model to multiple resources and considers a time dependent 'reimage' initiated by the defender, much like our full reset play of the defender described above. In addition [17], much like our own work, sets up a situation of asymmetric stealth in that the attacker can always tell when the defender has moved however the defender does not know when the attacker has compromised the resource but finds this out when he has probes the resource.

Having multiple resources which an attacker needs to compromise also models the situation of a moving target defence and a number of game theoretic works are devoted to other aspects of moving target defence including [3, 18]. Since these works are not directly related to our own work we do not discuss them here.

2 The Multi-Party FlipIt Model

Our basic multi-party FlipIt game, or FlipThem game, consists of a defender who is trying to protect against an attacker getting control of n different resources. It may help the reader to notice how at each point our game degenerates to the FlipIt game when $n = 1$.

At a given point in time the attacker will control a given threshold k of the resources. The attacker is deemed to be "in control", or have won, if k exceeds some value t. For example in a denial-of-service attack on a web site, the web-site may still be able to function even if $2/3$ of the servers are down, thus we will set $t = 2 \cdot n/3$. In the case of an attacker trying to disrupt a consensus building network protocol, i.e. an instantiation of the problem of Byzantine agreement, the value of t would be $n/3$. In the case of a multi-party computation protocol the threshold t would correspond to the underlying

threshold tolerated by the MPC protocol; e.g. $t = n/3$, $t = n/2$ or $t = n$. Note, in the case of MPC protocols, the ability of the defender to reset all resources is a common defence against mobile adversaries, and is thus related to what is called proactive security in the MPC community [11].

The variable D_B is the multiplicative factor of the defender's benefit (i.e. the benefit obtained per unit time), the same for the attacker's A_B. The values are potentially distinct, since the defender could gain more (or less) than the attacker for being in control of the system for an amount of time. The values D_c and A_c are respectively the defender and attacker's cost per action they perform. We set $d = \frac{D_c}{D_B}$ to be the ratio of the defender's cost and benefit. Similarly for the attacker, $a = \frac{A_c}{A_B}$. We then consider the ratio $\rho = \frac{a}{d} = \frac{A_c \cdot D_B}{A_B \cdot D_C}$. Much of our analysis will depend on whether ρ is large or small; which itself depends on the relative ratios of the benefit/costs of the attacker and defender. With each application scenario being different. A game where the costs are normalized in this way we shall call a "normalized game".

For each time period for which the attacker obtains control of t or more of the resources it obtains a given benefit, whereas for each time period that he does not have such control the defender obtains a benefit. In the normalized game we assume the attacker's benefit lies in $[0, 1]$ and is the proportion of time that he controls the resource; whilst the defenders benefit is the proportion of time in which they control the resource. Thus in the normalized game the benefits always sum to one.

In all games the utility for the attacker is their benefit minus their cost of playing (i.e. the cost of pushing the buttons), with the utility for the defender obtained in the same manner. Therefore, the game is non-zero sum. The attacker (resp. defenders) goal is to derive a strategy which maximises their respective utility.

In one basic normalised "Single Reset" game the defender has a set of n buttons; there is one button on each resource which when pressed will return that resource to the defenders control, or do nothing if the resource is already under the defenders control. Pressing the resource's button costs the defender a given value, which in the normalized game is the value d. In another normalised "Full Reset" game addition there is a "master button" which simultaneously returns all resources to the defenders control. Pressing the master button costs the defender a value which we shall denote by D_n, the value of which depends on n, the number of resources. The reason for having a master button is to capture the case when resetting the entire system in one go is simpler than resetting each resource individually. In particular we assume that $d \leq D_n$. To simplify our games we assume that the defender does not have access to the master button and the individual resource buttons in a single game. This property could be relaxed which would result in a much more complex analysis than that given here.

The attacker has a set of n buttons, one for each resource. When the attacker presses a resources button it will allow the adversary control of that resource, or again do nothing if the resource is already under the attackers control. The cost to the attacker of pressing one of its buttons is a in the normalized game.

As can be inferred from the above discussion we do not assume that the defender knows whether it controls a resource, nor do we assume that an attacker knows whether it controls a resource at a given time point. This situation is called the two-way stealthy

situation, if we assume a defender is not stealthy (but the attacker is) we are said to be in a one-way stealthy situation.

Throughout the paper we model a number of games. We denote $\text{FlipThem}^{\mathcal{R}}_{\epsilon}(n, t, d, \rho)$ to be the game of partial threshold FlipThem. By abuse of notation we also think of $\text{FlipThem}^{\mathcal{R}}_{\epsilon}(n, t, d, \rho)$ as a function which returns all the rates of play strategy pairs for the defender and attacker that are Nash Equilibria where $\mathcal{R} \in \{\mathcal{F}, \mathcal{S}\}$. Here we denote by $\mathcal{F}$ the full reset game and $\mathcal{S}$ the single reset game, both to be described in detail in later sections. The variables n, t, d ρ and ϵ denote the number of resources, the threshold, the defender's cost of play, the ratio between the attacker's and defender's cost and the lowest rate of play in the defender's strategy space $(\epsilon, \infty]$ respectively. Having $\epsilon > 0$ recognises the fact that the defender will never actually set the reset rate to 0. It also ensures that the benefit functions are well defined for all valid attacker-defender strategy pairs. We will not treat the choice of our ϵ to be strategic, it will be a very small number, close to zero to represent that even when the attacker has given up (plays a rate of zero) the defender will not.

We also use a function $\text{Opt}^{\mathcal{R}}_{N,\epsilon}(d, \mathcal{T}, \rho)$ to answer the following question: Given the ratio ρ of costs of play between the attacker and defender and a limit N for the number of resources the defender can own, what is the best set up for the defender in order to maximise their benefit function? The function $\text{Opt}^{\mathcal{R}}_{N,\epsilon}(d, \mathcal{T}, \rho)$ plays the first game $\text{FlipThem}^{\mathcal{R}}_{\epsilon}(n, t, d, \rho)$ for all n and all t subject to some constraint space $\mathcal{T}$[2]. The function $\text{Opt}^{\mathcal{R}}_{N,\epsilon}(d, \mathcal{T}, \rho)$ then finds the values of n and t which produce the greatest possible benefit for the defender.

3 Obtaining Nash Equilibria in Continuous Time for a Stochastic Process

In this section we analyse various different cases of our basic game $\text{FlipThem}^{\mathcal{R}}_{\epsilon}(n, t, d, \rho)$. To explain the basic analysis techniques in a simple example; we first examine the game $\text{FlipThem}^{\mathcal{F}}_{0}(n, n, d, \rho)$. In this game the defender can perform a full reset and the attacker is trying to compromise all n servers (i.e. the full threshold case). We also, again for initial simplicity and exposition purposes, assume that the defender could decide not to play, i.e. $\epsilon = 0$. A moments thought will reveal in practice that such a strategy is not realistic. In the later sub-sections we remove these two simplifying assumptions and examine other cases. In particular in Section 3.3 when we consider defender performing single resets, the analysis becomes more complex.

3.1 Simple Example, $\text{FlipThem}^{\mathcal{F}}_{0}(n, n, d, \rho)$: Full Threshold, Full Reset

We first consider a simple example of our framework in which the time an attacker takes to successfully compromise an individual resource follows an exponential distribution with rate λ, and the defender performs a full reset, and thus regains control of all resources, at intervals with lengths given by an exponential distribution with rate μ. An alternative description is that individual resources are compromised on at the arrival

[2] For example $t \leq n$, or $t \leq n/2$, or $n - t \geq B$ for some bound B.

times of a Poisson process with rate λ, and the state is reset at the arrival times of a Poisson process with rate μ.

In this context we think of the attacker as being stealthy, i.e. the defender does not know how many resources are compromised when he does a full reset. A moment's thought will also reveal that in this situation it makes no difference if the defender is stealthy or not; if the defender is not stealthy then the attacker will always pick an uncompromised resource to attack, whereas if the defender is stealthy then the attacker is more likely to compromise an uncompromised resource by picking one which he knows he controlled the longest time ago. Thus an attacker simply attacks each resource in turn, given some specific ordering.

We model the number of resources compromised by the attacker at time τ as a family of random variables $X = \{X(\tau) : \tau \geq 0\}$ in the finite space $S = \{0, \ldots, n\}$. Since both the defender and attacker follow memoryless strategies (with memoryless exponential random variables determining the times between changes of state) the process X is a continuous time Markov chain. Following the analysis of continuous time Markov chains in Grimmet et al. [5], for such a process there exists an $|S| \times |S|$ generator matrix G with entries $\{g_{ij} : i, j \in S\}$ such that

$$\Pr[X(\tau + h) = j \mid X(\tau) = i] = \begin{cases} 1 + g_{ii} \cdot h + o(h), & \text{if } j = i, \\ g_{ij} \cdot h + o(h), & \text{if } j \neq i. \end{cases}$$

The generator matrix G for continuous time Markov chains replaces the transition matrix P for discrete time Markov chains; entry g_{ij} for $i \neq j$ is the "rate" of transition from state i to state j. Summing equation (3.1) over j implies that $\sum_{j \in S} g_{ij} = 0$, so that $g_{ii} = -\sum_{j \neq i} g_{ij} \leq 0$. Basic theory [5] tells us that when the chain arrives in state i it remains there for an amount of time following a Exponential$(-g_{ii})$ distribution, then jumps to state $j \neq i$ with probability $-g_{ij}/g_{ii}$.

Considering our specific example with the defender using full reset, we can consider our model as a "birth-reset process" (by analogy with a "birth–death process") in which

$$\Pr[X(\tau + h) = j \mid X(\tau) = i] = \begin{cases} \lambda \cdot h + o(h), & \text{if } j = i + 1, \\ \mu \cdot h + o(h), & \text{if } j = 0, \\ 1 - (\lambda + \mu) \cdot h + o(h), & \text{if } j = i, \\ o(h), & \text{otherwise.} \end{cases}$$

Thus, $g_{i0} = \mu$, $g_{i,i+1} = \lambda$, $g_{ii} = -(\mu + \lambda)$ and $g_{ij} = 0$ otherwise. From this the generator matrix can be constructed:

$$G = \begin{pmatrix} -\lambda & \lambda & 0 & 0 \ldots & 0 & 0 \\ \mu & -(\mu+\lambda) & \lambda & 0 \ldots & 0 & 0 \\ \mu & 0 & -(\mu+\lambda) & \lambda \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \ddots & \vdots & \vdots \\ \mu & 0 & 0 & 0 \ldots & -(\mu+\lambda) & \lambda \\ \mu & 0 & 0 & 0 \ldots & 0 & -\mu \end{pmatrix}.$$

Thus when the state is $i \in \{1, \ldots, n-1\}$ the system will jump to either state $i+1$ with probability $\lambda/(\lambda + \mu)$ (when the attacker compromises another resource before reset

occurs) or to state 0 with probability $\mu/(\lambda+\mu)$ (when the reset occurs before another resource is compromised). Clearly the chain is never going to settle in one state; it will continue to randomly fluctuate between various states depending on the rates of play μ and λ. However further theory [5] indicates that the long run proportion of time the system spends in each state is given by the stationary distribution, a row vector $\pi = (\pi_0, \ldots, \pi_n)$ such that $\pi G = 0$ and $\sum_{i=0}^{n} \pi_i = 1$.

Using our specific generator matrix G it can be shown that

$$\pi = \left(\frac{\mu}{\mu+\lambda}, \frac{\mu \cdot \lambda}{(\mu+\lambda)^2}, \ldots, \frac{\mu \cdot \lambda^{n-1}}{(\mu+\lambda)^n}, \frac{\lambda^n}{(\mu+\lambda)^n} \right). \tag{1}$$

This tells us the proportion of time spent in each state. We therefore obtain the benefit functions of

$$\beta'_D(\mu, \lambda) = D_B \cdot (1 - \pi_n) - D_c \cdot \mu$$

and

$$\beta'_A(\mu, \lambda) = A_B \cdot \pi_n - A_c \cdot \lambda$$

where β'_D is the benefit function of the defender and β'_A is the benefit function of the attacker. We can then normalise β'_D and β'_A such that

$$\beta_D(\mu, \lambda) = \frac{\beta'_D}{D_B} = 1 - \pi_n - d \cdot \mu = 1 - \frac{\lambda^n}{(\mu+\lambda)^n} - d \cdot \mu$$

and

$$\beta_A(\mu, \lambda) = \frac{\beta'_A}{A_B} = \pi_n - a \cdot \lambda = \frac{\lambda^n}{(\mu+\lambda)^n} - a \cdot \lambda,$$

where β_D is the normalized benefit function of the defender and β_A is the normalized benefit function of the attacker.

Recall that in this model, when the defender plays he is resetting all resources at once. Therefore, the normalized cost of the defenders move d is likely to depend on n, the number of resources. We represent this by setting $d = D_n$.

Using the stationary distribution described above the benefit functions for the normalized game are

$$\beta_D(\mu, \lambda) = 1 - \frac{\lambda^n}{(\mu+\lambda)^n} - D_n \cdot \mu \quad \text{and} \quad \beta_A(\mu, \lambda) = \frac{\lambda^n}{(\mu+\lambda)^n} - a \cdot \lambda. \tag{2}$$

We are assuming that both players are rational, in that they are both interested in maximising their benefit functions, and will therefore choose a rate (λ or μ) to maximise their benefit given the behaviour of their opponent. A pair of rates at which each player is playing optimally against the other is called a Nash equilibrium [10]. At such a point neither player can increase their benefit by changing their rate; we are looking for pairs (λ^*, μ^*) such that

$$\beta_D(\mu^*, \lambda^*) = \max_{\mu \in R_+} \beta_D(\mu, \lambda^*) \quad \text{and} \quad \beta_A(\mu^*, \lambda^*) = \max_{\lambda \in R_+} \beta_A(\mu^*, \lambda).$$

Note that $\mu^* = \lambda^* = 0$ is an equilibrium of the game defined by equations in (2). This is an artefact of assuming the existence of a unique distribution for all μ, λ, where as

when $\lambda = \mu = 0$ the Markov chain never makes any transitions. In later sections we will bound μ below to remove this solution and for now we will search for non-trivial solutions.

Differentiating the defender's benefit function β_D with respect to μ and solving for μ gives at most one non-negative real solution, given by

$$\hat{\mu}(\lambda) = \sqrt[n+1]{\frac{n\lambda^n}{D_n}} - \lambda$$

If $\lambda < \frac{n}{D_n}$ then this is positive, and checking the second derivative confirms this corresponds to a maximum. If $\lambda \geq \frac{n}{D_n}$ then $\frac{\partial \beta_D}{\partial \mu} < 0$ for all $\mu \geq 0$ and so the optimal rate for the defender is $\mu = 0$. Hence the best response of the defender is given by

$$\hat{\mu}(\lambda) = \begin{cases} \sqrt[n+1]{\frac{n\lambda^n}{D_n}} - \lambda & \text{if } \lambda < \frac{n}{D_n} \\ 0 & \text{if } \lambda \geq \frac{n}{D_n}. \end{cases}$$

We now calculate

$$\frac{\partial \beta_A}{\partial \lambda} = \frac{n \cdot \mu \cdot \lambda^{n-1}}{(\mu + \lambda)^{n+1}} - a.$$

A closed form solution for λ which equates this to 0 is not easy to calculate directly. However, plugging in $\hat{\mu}(\lambda^*)$ we see that λ^* must be either 0 or satisfy

$$\frac{n \cdot \hat{\mu}(\lambda^*) \cdot (\lambda^*)^{n-1}}{(\hat{\mu}(\lambda^*) + \lambda^*)^{n+1}} - a = 0. \tag{3}$$

If it were the case that $\lambda^* \geq \frac{n}{D_n}$ then $\hat{\mu}(\lambda^*) = 0$ and there are no solutions to this equation. Note that this indicates that no equilibrium exists when the attacker's rate is too high — the intuition for this is if the attacker's rate is sufficiently high, the defender ceases to defend, and thus the attacker can do just as well by reducing their rate. Thus at any equilibrium we must have $\lambda^* < \frac{n}{D_n}$, and therefore $\mu^* = \hat{\mu}(\lambda^*) = \sqrt[n+1]{\frac{n(\lambda^*)^n}{D_n}}$. Plugging this back into equation (3) we see that either

$$\lambda^* = \frac{n \cdot D_n^n}{(D_n + a)^{n+1}}, \quad \mu^* = \hat{\mu}(\lambda^*) = \frac{n \cdot a \cdot D_n^{n-1}}{(D_n + a)^{n+1}}, \tag{4}$$

or $\mu^* = \lambda^* = 0$. The non-zero solution will only correspond to a Nash equilibrium if $\beta_A(\mu^*, \lambda^*) \geq \beta_A(\mu^*, 0) = 0$, since otherwise λ^* is not a best response against μ^*. Note that this is the case if

$$0 < \frac{(\lambda^*)^n}{(\mu^* + \lambda^*)^n} - a \cdot \lambda^* = \frac{(D_n)^n}{(D_n + a)^{n+1}}(D_n + a \cdot (1 - n))$$

i.e. if $a/D_N < 1/(n-1)$.

In the game $\text{FlipThem}_0^{\mathcal{F}}(n, n, D_n, \rho)$ we have defined ρ to be the ratio between the attacker and defender's costs, so that $\rho = a/D_n$. Therefore, the game $\text{FlipThem}_0^{\mathcal{F}}(n, n, D_n, \rho)$

returns the list $\{(0,0)\}$ for all $\rho > 1/(n-1)$. If $\rho < 1/(n-1)$ we have a further equilibrium (μ^*, λ^*) such that the game returns the list $\{(0,0),(\mu^*,\lambda^*)\}$ where

$$\mu^* = \frac{n \cdot \rho}{D_n \cdot (1+\rho)^{n+1}}, \quad \lambda^* = \frac{n}{D_n \cdot (1+\rho)^{n+1}} = \mu^*/\rho.$$

The attacker's cost per move is independent of n, which implies that the defender will be successful, assuming $\frac{D_n}{n-1}$ is a decreasing function of n, as long as n is large enough. Thus for the defender to always win we require the cost of a full reset to be a sublinear function of the number of resources.

In the case of resetting a cloud or web service this might be a reasonable assumption, but in the case of requiring n users to reset their passwords it is likely that the cost is a superlinear function as opposed to sublinear due to the social cost in needing to implement such a password policy.

3.2 FlipThem$^{\mathcal{F}}_{\epsilon}(n, t, d, \rho)$: (n,t)-Threshold, Full Reset

We now generalize the previous easy case to the threshold case FlipThem$^{\mathcal{F}}_{\epsilon}(\ n, t, d, \rho)$, i.e. we treat the number of servers which the attacker has to compromise as a parameter t, and in addition we bound the defenders strategy away from zero. Thus the defender not playing at all is not considered a valid strategy[3]. Much of the prior analysis carries through, since we are still assuming the defender performs a full reset on his turn. Thus the stationary distribution is once more,

$$\pi = \left(\frac{\mu}{\mu+\lambda}, \ldots, \frac{\mu \cdot \lambda^{k-1}}{(\mu+\lambda)^k}, \ldots, \frac{\mu \cdot \lambda^{n-1}}{(\mu+\lambda)^n}, \frac{\lambda^n}{(\mu+\lambda)^n} \right).$$

The (normalized) benefit functions are now derived from the ratio of times which the attacker has compromised at least t resources, which simplifies due to the formula for geometric series:

$$\begin{aligned} \beta_D(\mu,\lambda) &= 1 - \frac{\lambda^n}{(\mu+\lambda)^n} - \sum_{i=t}^{n-1} \frac{\mu \cdot \lambda^i}{(\mu+\lambda)^{i+1}} - D_n \cdot \mu \\ &= 1 - \frac{\lambda^t}{(\mu+\lambda)^t} - D_n \cdot \mu. \end{aligned}$$

Using the same analysis, the attacker's benefit is $\beta_A(\mu,\lambda) = \frac{\lambda^t}{(\mu+\lambda)^t} - a \cdot \lambda$. Note that these benefit functions are identical to those in the full threshold case of the previous section, but with n replaced by t. If we were still considering the lower bound for the defender's rate of play ϵ to be zero the conclusions would be as before, but with the modification that we use t instead of n. Since we are now considering the more realistic assumption that $\epsilon > 0$ the analysis gets slightly more involved, but remains similar to that above. In particular

$$\beta_D(\mu,\lambda) = 1 - \left(\frac{\lambda}{\lambda+\mu}\right)^t - D_n \cdot \mu, \quad \text{and} \quad \frac{\partial \beta_D}{\partial \mu} = \frac{t \cdot \lambda^t}{(\lambda+\mu)^{t+1}} - D_n.$$

[3] Of course if the attacker decides not to play that is considered a good thing.

This derivative is decreasing in μ, and 0 at $\lambda \cdot \left[\left(\frac{t}{\lambda \cdot D_n} \right)^{\frac{1}{t+1}} - 1 \right]$. It follows immediately that β_D is a unimodal function of μ, so that the maximising μ value in $[\epsilon, \infty)$ is given by

$$\hat{\mu}(\lambda) = \min \left\{ \epsilon, \lambda \cdot \left[\left(\frac{t}{\lambda \cdot D_n} \right)^{\frac{1}{t+1}} - 1 \right] \right\}. \tag{5}$$

As above, we have that

$$\beta_A(\mu, \lambda) = \left(\frac{\lambda}{\mu + \lambda} \right)^t - a \cdot \lambda \quad \text{and} \quad \frac{\partial \beta_A}{\partial \lambda} = \frac{t \cdot \mu \cdot \lambda^{t-1}}{(\lambda + \mu)^{t+1}} - a. \tag{6}$$

Thus for a particular value of μ the maximising λ must either be 0 or be a root of the derivative. However, explicitly solving for λ does not appear to be possible, but we note that

$$\frac{\partial^2 \beta_A}{\partial \lambda^2} = \frac{t \cdot \mu \cdot \lambda^{t-2}}{(\lambda + \mu)^{t+2}} \cdot [\mu \cdot (t-1) - 2 \cdot \lambda]$$

so that the first derivative, $\frac{\partial \beta_A}{\partial \lambda}$, is increasing when $\lambda < \mu \cdot (t-1)/2$ then decreasing. Since $\frac{\partial \beta_A}{\partial \lambda}$ is equal to $-a$ when $\lambda = 0$ and asymptotes to $-a$ as $\lambda \to \infty$ we have the derivative increasing from $-a$ to a maximum when $\lambda = \mu \cdot (t-1)/2$ then decreasing back to $-a$. The maximal value of $\frac{\partial \beta_A}{\partial \lambda}$ is given by

$$\frac{4 \cdot t \cdot (t-1)^{t-1}}{\mu \cdot (t+1)^{t+1}} - a, \tag{7}$$

which is positive only if μ is sufficiently small. As a function of λ, β_A therefore initially decreases (from 0), has a period of increase only if μ is sufficiently small, then decreases again. It follows that β_A has at most one non-zero maximum, which occurs in the region $(\mu \cdot (t-1)/2, \infty)$ once the derivative is decreasing, and this fixed point maximises $\beta_A(\mu, \lambda)$ on $\lambda \in [0, \infty)$ if and only if $\beta_A(\mu, \lambda) > 0$; otherwise the best response must be $\lambda = 0$. We use these insights to explore Nash equilibria directly. First consider the existence of a Nash equilibrium (μ^*, λ^*) with $\mu^* > \epsilon$. Note that if λ^* were equal to 0 then this would force $\mu^* = \epsilon$, so it must be the case that $\mu^* = \hat{\mu}(\lambda^*)$ and $\frac{\partial \beta_A}{\partial \lambda}(\mu^*, \lambda^*) = 0$. It follows from (5) and (6) that

$$a = \frac{t \cdot \mu^* \cdot {\lambda^*}^{t-1}}{(\lambda^* + \mu^*)^{t+1}} = D_n \cdot \left[\left(\frac{t}{\lambda^* \cdot D_n} \right)^{\frac{1}{t+1}} - 1 \right]$$

and hence

$$\lambda^* = \frac{t}{D_n \cdot (1+\rho)^{t+1}}, \qquad \mu^* = \frac{t \cdot \rho}{D_n \cdot (1+\rho)^{t+1}}. \tag{8}$$

We have checked necessary conditions so far, but have still not verified that this λ^* does correspond to a maximum of β_A. As observed above, the necessary and sufficient condition is that

$$0 < \beta_A(\mu^*, \lambda^*) = \frac{1 + \rho - \rho \cdot t}{(1+\rho)^{t+1}}.$$

Thus an equilibrium of this form exists when

$$\rho < \frac{1}{t-1} \quad \text{and} \quad \mu^* = \frac{t \cdot \rho}{D_n \cdot (1+\rho)^{t+1}} > \epsilon.$$

Therefore, if the ratio ρ of the attacker's cost and defender's cost is less than $\frac{1}{t-1}$ then the game $\text{FlipThem}^{\mathcal{F}}_{\epsilon}(n, t, d, \rho)$ returns the list consisting of two pairs, the trivial equilibrium of no play (from the attacker, the defender plays at minimal rate ϵ) and an equilibrium at

$$\mu^* = \frac{t \cdot \rho}{D_n \cdot (1+\rho)^{t+1}}, \quad \lambda^* = \frac{t}{D_n \cdot (1+\rho)^{t+1}} = \mu^*/\rho.$$

Note that if the maximal value of the derivative of β_A is non-positive then no stationary point of β_A exists, and so λ will be 0. By removing all local maxima of the attacker's payoff function we really would expect the attacker to just stop playing; i.e. this would be the perfect defenders strategy. From (7) we see that by taking

$$\epsilon \geq \frac{4 \cdot t \cdot (t-1)^{t-1}}{a \cdot (t+1)^{t+1}} \tag{9}$$

we can ensure there is only the trivial equilibrium. Note that a simpler lower bound on ϵ, which trivially implies the one above, is to take $\epsilon \geq \frac{4}{a \cdot (t+1)}$. Note that choosing a sufficiently high ϵ in this way is very conservative. The rate of decrease of β_A is $-a$ at $\lambda = 0$ and as $\lambda \to \infty$, so by insisting there is no local maximum at all we ensure β_A stays well away from 0.

Picking ϵ to force out the attacker only makes sense if the defender's benefit is actually maximised. It might be the case that stopping the attacker completely is not economically viable. Therefore, in such a case ϵ should be chosen to be very small, close to zero and the other equilibria in equation (8) should be used; implying that μ^* is less than the right hand side of equation (9). Thus an expected amount of attacker success may be tolerated if completely eliminating such success comes at too much of a price. Recall our function $\text{Opt}^{\mathcal{F}}_{N,\mathcal{T},\epsilon}(d, \rho)$. If we fix $\epsilon = 0.01/d$ and set $\mathcal{T} = \{t \leq n\}$, and run this programmatically for ρ from 0 to 1, Fig. 1 shows the smallest $n \leq N$ that maximises the defenders benefit for various N. Recall that the attacker will not play if $\rho > 1/t - 1$, meaning that as ρ increases the level of threshold decreases and therefore the number of servers required decrease. The optimum defender's benefit occurring when $t = n$. This explains the step down in Fig. 1.

We end this section by examining the classic case of a threshold situation in which the required threshold is a constant fraction of the total number of resources. Suppose we have $t = \gamma \cdot n$ for some constant $\gamma \in (0, 1]$. We have shown that the attacker will not play if $\frac{a \cdot \rho}{D_n} \geq \frac{1}{t-1} = \frac{1}{\gamma \cdot n - 1}$. As expected we see that if the attacker needs to compromise fewer resources, then the attacker's cost per resource needs to be greater for them not to play. It is intuitively obvious that the smaller the threshold the more likely the attacker will play (and succeed).

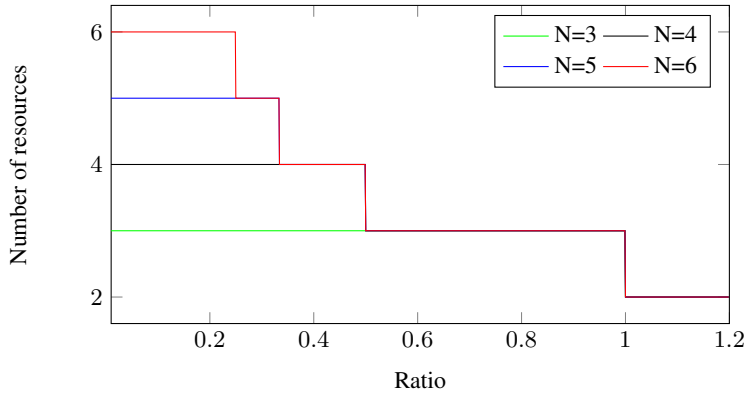

Fig. 1. Number of resources used by the defender to maximise his benefit given a specific ρ

3.3 FlipThem$^{\mathcal{S}}_{\epsilon}(n, t, d, \rho)$: (n,t)-Threshold, Single Reset

So far we have set up the model such that the defender can reset the whole system regaining full control whereas the attacker compromises each resource individually. We now consider the game FlipThem$^{\mathcal{S}}_{\epsilon}(n, t, d, \rho)$. The defender can reset a single machine at any specific time. Consider the situation at any time point where the number of resources compromised is k out of n. Assume the defender is going to reset a resource. There are multiple strategies they could employ, they could pick a resource which they have not reset recently, or pick a random resource, or pick a resource in a given secret sequence. Here we will assume the players pick resources uniformly at random. Thus the probability of resetting a compromised resource is $\frac{k}{n}$, and that of wastefully resetting a non-compromised resource $1 - \frac{k}{n}$. Letting the defender's and attacker's rate of play be μ and λ respectively, it is not hard to see that our generating matrix now becomes

$$G = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots & 0 & 0 & 0 \\ \frac{\mu}{n} & -\frac{(\mu+(n-1)\cdot\lambda)}{n} & \frac{(n-1)\cdot\lambda}{n} & 0 & \dots & 0 & 0 & 0 \\ 0 & \frac{2\cdot\mu}{n} & -\frac{(2\cdot\mu+(n-2)\cdot\lambda)}{n} & \frac{(n-2)\cdot\lambda}{n} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{(n-1)\cdot\mu}{n} & -\frac{((n-1)\cdot\mu+\lambda)}{n} & \frac{\lambda}{n} \\ 0 & 0 & 0 & 0 & \dots & 0 & \mu & -\mu \end{pmatrix}$$

We then solve for the stationary distribution $\pi = (\pi_0, \pi_1, \dots, \pi_{n-1}, \pi_n)$, by solving $\pi G = 0$, and it can be shown by induction that

$$\pi_k = \frac{n! \cdot \lambda^k \cdot \pi_0}{(n-k)! \cdot k! \cdot \mu^k} = \frac{\binom{n}{k} \cdot \lambda^k \cdot \pi_0}{\mu^k}.$$

Recall, that we also need to utilize the constraint $\sum_{i=0}^{n} \pi_i = 1$, which implies that we have $\pi_0 = \frac{\mu^n}{(\mu+\lambda)^n}$ so that we obtain the stationary distribution

$$\pi = \frac{1}{(\mu+\lambda)^n}\left(\mu^n, n\cdot\lambda\cdot\mu^{n-1}, \ldots, \binom{n}{k}\cdot\mu^{n-k}\cdot\lambda^k, \ldots, n\cdot\mu\cdot\lambda^{n-1}, \lambda^n\right).$$

Once again, this gives us the proportion of time spent in each state. We assume here that the costs and benefits have already been normalised and do not depend on n the number of resouces. Constructing these benefit functions gives

$$\beta_D(\mu,\lambda) = 1 - \sum_{i=t}^{n}\pi_i - d\cdot\mu = 1 - \frac{1}{(\mu+\lambda)^n}\cdot\sum_{i=t}^{n}\binom{n}{i}\cdot\mu^{n-i}\cdot\lambda^i - d\cdot\mu,$$

$$\beta_A(\mu,\lambda) = \sum_{i=t}^{n}\pi_i - a\cdot\lambda = \frac{1}{(\mu+\lambda)^n}\cdot\sum_{i=t}^{n}\binom{n}{i}\cdot\mu^{n-i}\cdot\lambda^i - a\cdot\lambda$$

We want to find the Nash Equilibria for these benefit functions. A point at which neither player can increase their benefit by changing their rate. We want to find pairs (μ^*, λ^*) such that

$$\beta_D(\mu^*,\lambda^*) = \max_{\mu\in(\epsilon,\infty)}\beta_D(\mu,\lambda^*) \quad \text{and} \quad \beta_A(\mu^*,\lambda^*) = \max_{\lambda\in R_+}\beta_A(\mu^*,\lambda),$$

where ϵ is the lowest rate we can expect the defender to play in order to ensure the stationary distributions and hence benefit functions are well defined for all valid (μ,λ). Differentiating the defender's and attacker's functions with respect to μ and λ respectively gives,

$$\frac{\partial\beta_D}{\partial\mu} = \frac{n!\cdot\mu^{n-t}\cdot\lambda^t}{(t-1)!\cdot(n-t)!\cdot(\mu+\lambda)^{n+1}} - d, \tag{10}$$

$$\frac{\partial\beta_A}{\partial\lambda} = \frac{n!\cdot\mu^{n-t+1}\cdot\lambda^{t-1}}{(t-1)!\cdot(n-t)!\cdot(\mu+\lambda)^{n+1}} - a. \tag{11}$$

Closed form solutions for μ and λ which equate to 0 are not easy to calculate directly. The second derivative of the attackers benefit with respect to λ is

$$\frac{n!\cdot\mu^{n-t+1}\cdot\lambda^{t-2}}{(t-1)!\cdot(n-t)!\cdot(\mu+\lambda)^{n+2}}\cdot[\mu\cdot(t-1) - \lambda\cdot(n+2-t)].$$

Thus, $\frac{\partial\beta_A}{\partial\lambda}$ is increasing when

$$\lambda < \frac{\mu\cdot(t-1)}{n+2-t},$$

then decreasing. Since $\frac{\partial\beta_A}{\partial\lambda}$ is $-a$ at $\lambda = 0$ and asymptotes to $-a$ as $\lambda\to\infty$ we have the derivative increasing from $-a$ to a maximum when $\lambda = \frac{\mu\cdot(t-1)}{n+2-t}$ and then decreasing back to $-a$. The maximal value of $\frac{\partial\beta_A}{\partial\lambda}$ is given by

$$\frac{n!\cdot(t-1)^{t-1}}{t^{n+1}\cdot(n+2-t)^{t-2}\cdot\mu} - a \tag{12}$$

which is positive only if μ is sufficiently small. As a function of λ, β_A therefore initially decreases (from 0), has a period of increase only if μ is sufficiently small, then decreases again. It follows that β_A has at most one non-zero maximum which occurs in the region

$$\left(\frac{\mu(t-1)}{n+2-t}, \infty\right)$$

once the derivative is decreasing, and this fixed point maximises $\beta_A(\mu, \lambda)$ on $\lambda \in [0, \infty)$ if and only if $\beta_A(\mu, \lambda) > 0$; otherwise the best response must be $\lambda = 0$. First, like the full reset case, we consider the existence of a Nash Equilibrium (μ, λ) with $\mu > \epsilon$. Since both derivatives (10) and (11) are hard to solve analytically for general n, we used a numerical method utilizing the Maple algebra system to solve for a specific n. The method for solving starts with defining the benefit functions in terms of μ and λ, we then differentiate the derivatives as above and solve for μ and λ for the defender and attacker, respectively. This provides 2 generic solutions of the form

$$\hat{\mu}(\lambda) = \text{RootOf}(f(\lambda)) \quad \text{and} \quad \hat{\lambda}(\mu) = \text{RootOf}(g(\mu))$$

where f and g are polynomials. We then put these solutions back into the derivatives to give

$$\frac{\partial \beta_D(\mu, \hat{\lambda}(\mu))}{\partial \mu} \quad \text{and} \quad \frac{\partial \beta_A(\hat{\mu}(\lambda), \lambda)}{\partial \lambda}$$

Solving these with respect to μ and λ respectively gives solutions for μ^* and λ^* with respect to the costs d and a. From this we can consider the ratio $\rho = \frac{a}{d}$ between the attacker's and defender's costs of play. A table can be constructed to show the ratios at which both the defender and attacker will and won't play for various ρ. Recall that even if the attacker is not playing, the defender must still play at some rate ϵ in order to ensure control of the system. In order to calculate the defender's benefit given a specific ρ we must calculate the lowest rate of play for the defender when the attacker is not playing. From equation (12), $\frac{\partial \beta_A}{\partial \lambda}$ is never positive if

$$\mu > \frac{n! \cdot (t-1)^{t-1}}{t^{n+1} \cdot (n+2-t)^{t-2} \cdot a}$$

Meaning no stationary point exists for the attackers benefit. From this we can see that by taking

$$\epsilon \geq \frac{n! \cdot (t-1)^{t-1}}{t^{n+1} \cdot (n+2-t)^{t-2} \cdot a}$$

we can ensure there is no equilibrium with $\mu^* = \epsilon$ and $\lambda \neq 0$. Recall that $\rho = \frac{a}{d}$, so that

$$\epsilon \geq \frac{n! \cdot (t-1)^{t-1}}{t^{n+1} \cdot (n+2-t)^{t-2} \cdot \rho \cdot d}$$

This shows that if ρ is large enough, ϵ will be small meaning the likely strategy for the attacker will be no play, $\lambda = 0$. So the benefit for the defender will be

$$\beta_D(\epsilon, 0) = 1 - \epsilon \cdot d = 1 - \frac{n! \cdot (t-1)^{t-1}}{t^{n+1} \cdot (n+2-t)^{t-2} \cdot \rho}.$$

However, having ρ large enough to ensure ϵ is small enough is an unrealistic assumption and choosing ϵ like this becomes a strategic choice. As it was for the full reset case, it is also very conservative and could be expensive for the defender. We therefore fix our $\epsilon > 0$ to be very small, close to zero before the game. We now want to ask the following question: Given the costs of play for both defender and attacker and a limit N for the number of resources the defender can own, what is the best set up for the defender in order to maximise their benefit function? i.e. given ρ and N we are looking for the pairs such that

$$\beta_D^*(n^*, t^*) = \max_{n \leq N, t \leq n} \beta_D^*(n, t)$$

where $\beta_D^*(n,t) = \beta_D(\mu^*, \lambda^*)$ is the Nash equilibrium for the specific number of resources n and threshold t. Recall we defined this game to be $\text{Opt}^{\mathcal{S}}_{N,\epsilon}(d, \mathcal{T}, \rho)$. We turn to the method of numerical programming for this problem. Obviously, since the lowest rate of play ϵ for the defender is chosen arbitrarily before the game is played, if the equilibrium played is the trivial equilibrium then the defenders benefit is $\beta_D(\epsilon, 0) = 1 - \epsilon \cdot d$.

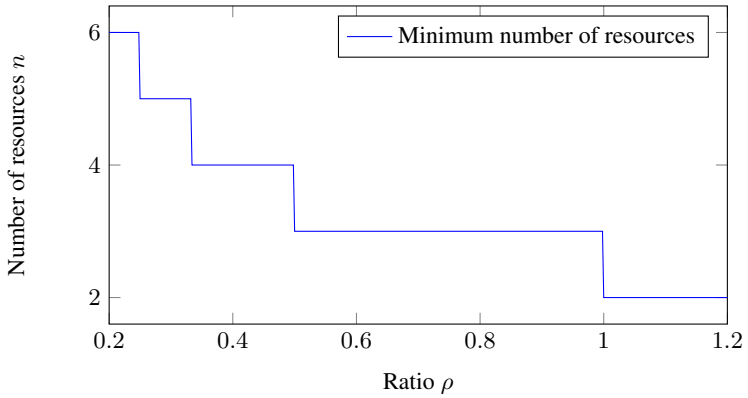

Fig. 2. Number of resources used by the defender to maximise his benefit given a specific ρ, for $\mathcal{T} = \{t \leq n\}$ and $N = 7$.

When running $\text{Opt}^{\mathcal{S}}_{N,\epsilon}(d, \mathcal{T}, \rho)$, each round of $\text{FlipThem}^{\mathcal{S}}_{\epsilon}(n, t, d, \rho)$ played has three possible outcomes.

- If ρ is so small the defender will not even play at the minimal rate ϵ.
- If ρ is "mid-size" the defender and attacker both play the non-trivial equilibrium (μ^*, λ^*).
- If ρ is large the attacker does not play and the trivial equilibrium $(\epsilon, 0)$ is played.

We experimentally examined two scenarios, both in which we fix $\epsilon = 0.01/d$. In the first scenario we take $\mathcal{T} = \{t \leq n\}$ and $N = 7$, in this case the function $\text{Opt}^{\mathcal{S}}_{N,\epsilon}(d, \mathcal{T}, \rho)$ outputs valid configurations for relatively small values of ρ, see Fig. 2. Interestingly the output best game for a maximum defenders benefit is always a full threshold game. In the second scenario we take $\mathcal{T} = \{t < n/2\}$, and again $N = 7$. The results are given

in Fig. 3. In this case small values of ρ result in games for which the defender will not play, for larger values of ρ we end up requiring more servers.

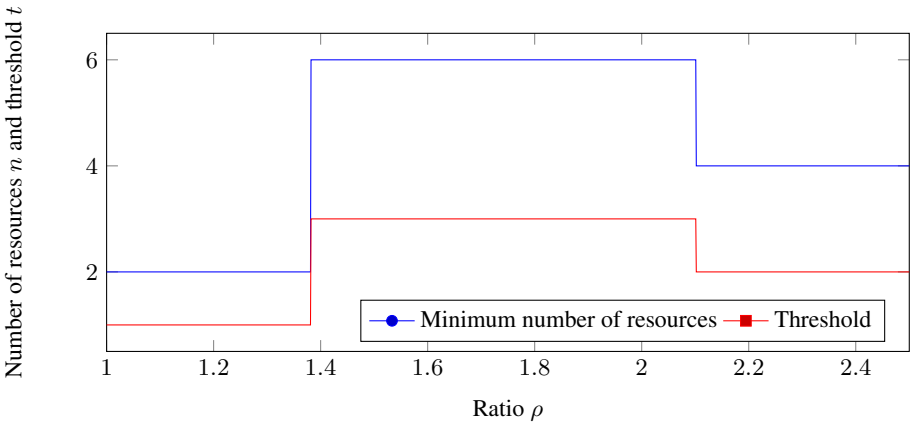

Fig. 3. Number of resources used by the defender to maximise his benefit given a specific ρ, for $\mathcal{T} = \{t < n/2\}$ and $N = 7$.

Acknowledgements

The second author was supported by a studentship from GCHQ. This work has been supported in part by ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO and by EPSRC via grant EP/I03126X.

References

1. H. S. Bedi, S. G. Shiva, and S. Roy. A game inspired defense mechanism against distributed denial of service attacks. *Security and Communication Networks*, 7(12):2389–2404, 2014.
2. K. D. Bowers, M. van Dijk, R. Griffin, A. Juels, A. Oprea, R. L. Rivest, and N. Triandopoulos. Defending against the unknown enemy: Applying flipit to system security. In Grossklags and Walrand [6], pages 248–263.
3. M. P. Collins. A cost-based mechanism for evaluating the effectiveness of moving target defenses. In Grossklags and Walrand [6], pages 221–233.
4. S. K. Das, C. Nita-Rotaru, and M. Kantarcioglu, editors. *Decision and Game Theory for Security - 4th International Conference, GameSec 2013, Fort Worth, TX, USA, November 11-12, 2013. Proceedings*, volume 8252 of *Lecture Notes in Computer Science*. Springer, 2013.
5. G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition edition, 2001.
6. J. Grossklags and J. C. Walrand, editors. *Decision and Game Theory for Security - Third International Conference, GameSec 2012, Budapest, Hungary, November 5-6, 2012. Proceedings*, volume 7638 of *Lecture Notes in Computer Science*. Springer, 2012.

7. A. Laszka, G. Horvath, M. Felegyhazi, and L. Buttyan. Flipthem: Modeling targeted attacks with flipit for multiple resources. In Poovendran and Saad [14], pages 175–194.
8. A. Laszka, B. Johnson, and J. Grossklags. Mitigation of targeted and non-targeted covert attacks as a timing game. In Das et al. [4], pages 175–191.
9. B. Z. Moayedi and M. A. Azgomi. A game theoretic framework for evaluation of the impacts of hackers diversity on security measures. *Rel. Eng. & Sys. Safety*, 99:45–54, 2012.
10. J. Nash. Non-cooperative games. *The Annals of Mathematics*, 54:286–295, 1951.
11. R. Ostrovsky and M. Yung. How to withstand mobile virus attacks (extended abstract). In L. Logrippo, editor, *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing, Montreal, Quebec, Canada, August 19-21, 1991*, pages 51–59. ACM, 1991.
12. E. Panaousis, A. Fielder, P. Malacaria, C. Hankin, and F. Smeraldi. Cybersecurity games and investments: A decision support approach. In Poovendran and Saad [14], pages 266–286.
13. V. Pham and C. Cid. Are we compromised? modelling security assessment games. In Grossklags and Walrand [6], pages 234–247.
14. R. Poovendran and W. Saad, editors. *Decision and Game Theory for Security - 5th International Conference, GameSec 2014, Los Angeles, CA, USA, November 6-7, 2014. Proceedings*, volume 8840 of *Lecture Notes in Computer Science*. Springer, 2014.
15. S. Roy, C. Ellis, S. G. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A survey of game theory as applied to network security. In *43rd Hawaii International International Conference on Systems Science (HICSS-43 2010), Proceedings, 5-8 January 2010, Koloa, Kauai, HI, USA*, pages 1–10. IEEE Computer Society, 2010.
16. M. van Dijk, A. Juels, A. Oprea, and R. L. Rivest. Flipit: The game of "stealthy takeover". *J. Cryptology*, 26(4):655–713, 2013.
17. M. P. Wellman and A. Prakash. Empirical game-theoretic analysis of an adaptive cyber-defense scenario (preliminary report). In Poovendran and Saad [14], pages 43–58.
18. Q. Zhu and T. Basar. Game-theoretic approach to feedback-driven multi-stage moving target defense. In Das et al. [4], pages 246–263.