Sufficiently Secure Peer-to-Peer Networks

Rupert Gatti[1], Stephen Lewis[2], Andy Ozment[2],
Thierry Rayna[1], and Andrei Serjantov[2]

[1]Faculty of Economics and Politics, University of Cambridge,
Austin Robinson Building, Sidgwick Avenue,
Cambridge CB3 9DD
{rupert.gatti,trayna}@econ.cam.ac.uk

[2]Computer Laboratory, University of Cambridge,
William Gates Building, 15 JJ Thomson Avenue,
Cambridge CB3 0FD
FirstName.LastName@cl.cam.ac.uk

Abstract

Threat models in computer security often consider a very powerful adversary. A more useful model may be to consider conflict in which both sides have economic considerations that limit the resources they are willing to devote to the conflict. This paper examines censorship resistance in a peer-to-peer network. A simple game theoretic model is examined and then elaborated to include multiple publishers, non-linear cost functions, and non-trivial search heuristics. In each elaboration, we examine the equilibrium behaviour of the censor and the publisher.

1 Introduction

The threat models employed in computer security research are often very powerful. The attacker commonly owns the entire network: he is able to delay, reorder, and modify messages arbitrarily. Indeed, the only restriction on the attacker is that he is unable to perform computationally infeasible tasks: breaking cryptography, finding collisions on hash functions, etc. When peer-to-peer systems came to be examined, the threat model employed in the literature was similar to the above [Dou02].

Here, we argue that peer-to-peer systems can be usefully considered with a more subtle threat model: participants in the system who have opposing interests can choose the level of resources they want to devote to 'fighting' each other. The attacker derives utility from thwarting a particular transaction, which could be file sharing, computation, document retrieval, etc. A 'good' peer derives utility from succeeding with a transaction. Given

Presented at the:
Workshop on Economics and Information Security. May 13-14, 2004: Minneapolis, MN, USA.

this scenario, it is possible to find the participants' strategies in equilibrium – or show that none exists.

If an attack is expensive, then the utility gained from attacking the network might be smaller than the cost of the attack. This means that for the network to be secure, it suffices to show that for any of the possible attacks on it, the associated cost is higher than the utility gained from it. Hence, a network merely needs to be *sufficiently secure* rather than secure in the (traditional) technical sense. For example, a peer-to-peer network designed for sharing family photos might be secure not because of the technical protection measures in use, but because there is no incentive to attack it.

This framework of incentive security allows us to describe more precisely the viability of a peer-to-peer network. The scenarios to which this framework can be applied include distributed computation [ACK+02], decentralized backup [HR02], censorship resistant systems [And96, Ser02, WRC00] and many others (even some terrorist organizations are peer-to-peer networks!). In this paper we focus on an attacker trying to censor a document in a censorship resistant system. The two opposing parties in this network are the publisher, who wishes his document to be available to other users on the network, and the attacker, who wants to remove it.

The basic assumptions of our model are as follows:

- The network consists of n nodes.
- The publisher distributes his document to d randomly chosen nodes.
- The attacker simultaneously attacks a fraction x of the n nodes (again, chosen randomly).
- If all the nodes where the publisher has placed his document are attacked, the attacker gains utility V_a.
- If one or more of the nodes the publisher has chosen remain uncorrupted, the publisher gains utility V_p.
- There are costs $c_a(x)$ and $c_p(d)$ associated with attacking and publishing.

We begin by analyzing the simple conflict between the publisher and the attacker. The publisher's goal is to ensure that at least one copy of his document exists on the system. The attacker's aim is to eliminate all copies of the document. Hence, the publisher controls the number of copies of the document he places on the system, d, while the attacker chooses the fraction of nodes to corrupt, x.

In Section 2, we consider the case where the attacker's cost function is linear in the number of nodes attacked (nx), and the publisher's cost function is linear in the number of nodes to which his document is published (d). In this case we find no Nash equilibria in pure strategies.

We then generalize the model in Section 3 by making the attacker's cost function non-linear in the number of nodes attacked: his cost is now proportional to $(nx)^\alpha$. With this generalization we are able to find the conditions for a Nash equilibrium in pure strategies to exist. In Section 3.3 we find the condition for the existence of more than one pure strategy Nash equilibrium.

The model is then developed further to cover the situation in which multiple publishers use the system to publishes their respective documents. In this case, the attacker's goal is to completely eliminate as many documents as possible. We finally introduce another agent into the system: the retriever. We develop a more complex model of search than that used previously, so that the attacker can derive utility from making it difficult (but not impossible) to retrieve a document from the system.

2 The Basic Model

Although game theoretic models have been used before in the analysis of peer-to-peer networks, for example in [GLBML01], they have not been used to consider the kind of conflict that occurs in a system providing censorship resistance.

In our model, the number of nodes in the network, n, is given exogenously. In other words, we are considering the behaviour of a publisher and an attacker in an existing network, such as Freenet [CSWH01] (for which n is around 10 000 nodes) or Free Haven [DFM01]. The publisher must decide on the number of these nodes, d, to which he wishes to deploy his document. To simplify the analysis, it is assumed that these d nodes are selected uniformly, at random, and with replacement from the set of all nodes. The attacker makes a simultaneous decision to corrupt a proportion of nodes x (nx nodes). These are also selected at random but without replacement. This is perfectly consistent with a censorship resistant system which provides anonymity to the servers which are storing the document, such as the one described in [Ser02].

The goal of the publisher is to ensure that at least one copy of his document is available on a node that has not been corrupted. Thus we assume that our censorship resistance scheme has "perfect searching" as a feature: if a single copy of the document exists in the system, it will be retrievable. An alternative model is proposed in Section 7. Note also that this requires a fairly high degree of resistance in the network: however many nodes the attacker compromises, the network will remain connected and provide perfect retrieval. This is, perhaps, more accurate in the case of of a terrorist organization than in the current designs of peer-to-peer networks! Conversely, the attacker aims to ensure that no copies of the document are available (i.e. that the d nodes to which the document has been sent have all been corrupted).

The probability that all the nodes to which the document is published are bad is x^d, so our expected utility functions for the publisher (EU_p) and the attacker (EU_a) are:

$$EU_p = V_p[1 - x^d] - c_p d \quad (1)$$
$$EU_a = V_a x^d - c_a n x \quad (2)$$

In this initial analysis our cost functions are linear in the number of nodes published to or attacked. Normalizing these utility functions such that $c_p = c_a = 1$, we get

$$EU_p = V_p[1 - x^d] - d \quad (3)$$
$$EU_a = V_a x^d - nx \quad (4)$$

Note that now V_a and V_p represent the ratio between the utility derived from succeeding in attack or publication, and the 'cost' (in utility terms) of performing the publication or attack.

The publisher has control over d, and the attacker has control over x, so we can now proceed towards finding any equilibria that exist in this situation. First we find the attacker's best response to any d chosen by the publisher. This is defined by the maximization problem

$$\max_{0 \leqslant x \leqslant 1} \left[V_a x^d - nx \right] \quad (5)$$

To find the value (or values) of x where $V_a x^d - nx$ is at a maximum, we take partial derivatives with respect to x.

$$\frac{\partial EU_a}{\partial x} = d V_a x^{(d-1)} - n \quad (6)$$
$$\frac{\partial^2 EU_a}{\partial x^2} = d(d-1) V_a x^{(d-2)} \quad (7)$$

First consider the case where $0 \leqslant x \leqslant 1$, $1 < d \leqslant n$ and $V_a > 0$. Here the second derivative (see Equation 7) is always positive, so in order to maximize his utility, the attacker must set x to either 0 or 1 (the bounds of the possible range).

We now consider the cases where $d = 0$ or $d = 1$. If d is 1, Equation 4 reduces to a linear function in x

$$EU_a = (V_a - n)x \quad (8)$$

Hence if $V_a/1 < 1$, the utility-maximizing choice for the attacker is to set b to 0; if $V_a/n > 1$, he should set x equal to 1. (In the case where $V_a/n = 1$, he is indifferent). When d is zero, it is clear that he should set x to 0 also (no copies of the document are deployed, so there is no reason to attack the network).

We have therefore shown that in this model, under the assumption that $V_a \neq n$, the attacker will always choose either to corrupt all the nodes in the network, or to corrupt none. If he chooses the former strategy, $U_a = V_a - nx$; if the latter, $U_a = 0$. The best strategy for the attacker is therefore to set

$$x = 0 \qquad \text{where } d = 0 \text{ or } V_a/n < 1 \tag{9}$$

$$x = 1 \qquad \text{otherwise} \tag{10}$$

It is now possible to consider the strategy of the publisher; this is simpler, because now only the best responses to $x = 0$ and $x = 1$ need to be considered. Under the assumption that the publisher will derive positive utility from successfully publishing a single copy of the document (i.e. $V_p > 1$), when $x = 0$ (i.e. the attacker chooses to corrupt no nodes), d should be set to 1. If, however, $x = 1$, the publisher should send out no copies (set $d = 0$), as there will be no hope of them 'surviving'.

We thus have a game between the publisher and the attacker with the payoff matrix is shown in Figure 1. The symbols A and P denote attack of the entire network and publishing of one copy of the document respectively. $\bar{A}$ and $\bar{P}$ represent not attacking and not publishing. Under the assumptions that $V_a/n > 1$ and $V_p > 1$, this game has no Nash equilibrium in pure strategies. If, however, we reverse the first constraint (i.e. $V_a/n < 1$), we find a Nash equilibrium where $d = 1$ and $x = 0$. This is the situation where it is never worth the attacker's while to attack the entire network, because the benefit he derives from the attack is smaller than the cost to him of attacking – recall the photo album example. (It is not useful to relax the second contraint: it should always be worthwhile for the publisher to publish at least one copy of his document.)

3 Non-linear cost of attack

We now consider the case where the cost of attack is non-linear in the number of nodes corrupted. Note that the condition $\alpha > 1$ ensures that the attacker gets decreasing returns to scale.

$$EU_p = V_p[1 - x^d] - d \tag{11}$$

$$EU_a = V_a x^d - (nx)^\alpha \tag{12}$$

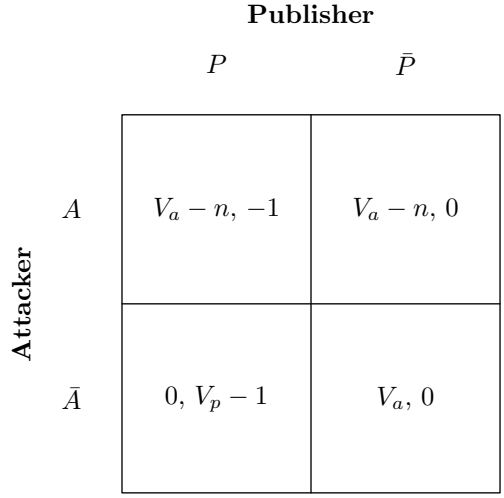

Figure 1: Payoff matrix for publish/censor game

Our first and second order conditions giving an interior solution for x ($0 < x < 1$) are

$$dV_a x^{(d-1)} - \alpha n^\alpha x^{\alpha-1} = 0 \quad (13)$$

$$dV_a(d-\alpha)x^{(d-2)} \leqslant 0 \quad (14)$$

The second order condition (14) has been simplified by assuming that the first order condition for x already holds. It is clear from (14) that an interior solution will be obtained when $d < \alpha$. We first consider, however, the case when $d > \alpha$.

3.1 $d > \alpha$

There are no Nash equilibria in pure strategies with $d > \alpha$. To see why, observe that the second derivative of the utility function (14) is always positive, so when the first derivative is equal to 0, the utility function has a minimum between $x = 0$ and $x = 1$. Hence, to maximize the utility, the attacker will set x to 0 or 1, to which the publisher's response will be to set d to 1 or 0 respectively. Observe that in both cases, $d < \alpha$.

3.2 $d < \alpha$

Where $d < \alpha$ it may be possible to get an interior solution: we know that there is an interior maximum of the attacker's expected utility function. The first and second derivatives of the publisher's expected utility are

$$\frac{\partial EU_p}{\partial d} = -V_p x^d \ln x - 1 \tag{15}$$

$$\frac{\partial^2 EU_p}{\partial d^2} = -V_p x^d (\ln x)^2 \tag{16}$$

The form of the conditions for equilibrium means that they are impossible to solve analytically. The second derivative given in (16) is always negative, however, so except for the special cases where its integer maximum is at $d = 0$ or $d = n$, the integer maximum will be at $d = k$ when

$$EU_p(k-1) < EU_p(k) \qquad \text{and} \tag{17}$$

$$EU_p(k+1) < EU_p(k) \tag{18}$$

We define x_k^* to be the attacker's best response at $d = k$ (solving (13) for x).

$$x_k^* = \left(\frac{\alpha n^\alpha}{dV_a}\right)^{1/(k-\alpha)} \tag{19}$$

A constraint on V_p can now be found that gives an equilibrium with $d = k$ and $x = x_k^*$. Figure 2 shows how the range of possible V_p giving pure strategy equilibria changes with increasing n.

$$\frac{1}{x^{k-1}(1-x)} < V_p < \frac{1}{x^k(1-x)} \tag{20}$$

With these constraints it is now possible to find an example of a pure strategy equilibrium. In a network with 1000 nodes and 2 copies of the publisher's document deployed, we set $\alpha = 3$ and $V_a = 3 \times 10^9$. The attacker's best strategy is to attack 2/3 of the network, and thus any V_p between 4.5 and 6.75 will give an equilibrium in pure strategies.

3.3 Multiple equilibria

It is also possible to find situations where there are multiple pure strategy equilibria for given parameters n, α, V_a and V_p. Intuitively, this is the case where the bound on V_p with $d = k$ overlaps with the bound on V_p with $d = k+1$. The constraint for this to hold is

$$\frac{1}{x_k^{*\,k}(1-x_k^*)} < \frac{1}{{x_{k+1}^*}^k(1-x_{k+1}^*)} \tag{21}$$

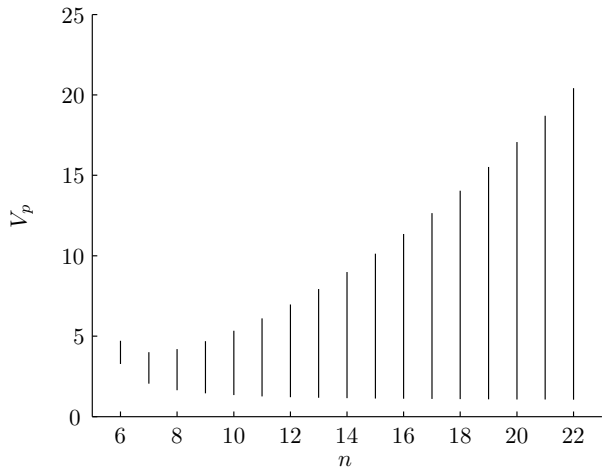

Figure 2: V_p ranges giving pure strategy equilibria ($V_a = 50$, $\alpha = 2$ and $d = 1$)

4 Mixed Strategies

The game between the attacker and the publisher (Figure 1) does not always have a Nash equilibrium in pure strategies; this is true even in the non-linear cost case treated in the previous section. We now look for a mixed strategy equilibrium for these cases, where the attacker either attacks the whole network or none of it, and the publisher publishes one copy of his document, or no copies.

In a mixed strategy equilibrium, each of the players plays one of the pure strategies with a certain probability. The equilibrium in this case is defined by a pair of probabilities (one for each player) such that the probability chosen by the first player is the best response to the probability chosen by the second player and vice versa.

We define λ_p, the probability of the publisher playing P (i.e. publishing one copy of his document) and λ_a, the probability of the attacker playing A (i.e. attacking and corrupting all the nodes). The expected utility for each player is then given by:

$$\begin{aligned} EU_a &= \lambda_a\lambda_p(V_a - n^\alpha) + \lambda_a(1-\lambda_p)(V_a - n^\alpha) + (1-\lambda_a)(1-\lambda_p)V_a \\ EU_p &= -\lambda_a\lambda_p + (1-\lambda_a)\lambda_p(V_p - 1) \end{aligned}$$

The first order conditions are:

$$\begin{aligned}\frac{\partial EU_a}{\partial \lambda_a} &= V_a \lambda p - n^\alpha = 0 \\ \frac{\partial EU_p}{\partial \lambda_p} &= -1 - V_p(\lambda_a - 1) = 0\end{aligned}$$

Solving these equations simultaneously, we get the following equilibrium:

$$\begin{aligned}\lambda_a &= \frac{V_p - 1}{V_p} \\ \lambda_p &= \frac{n^\alpha}{V_a}\end{aligned}$$

The first thing we notice is that the probability of the attack, λ_a, depends on how much the publisher values the dissemination of the document relative to the cost of publishing it. High utility attached to dissemination or low cost of publishing give high V_a, hence higher the probability of attack. Conversely, high cost of publishing or low utility attached to dissemination give low probability of attack (as the publisher is less likely to publish).

The probability of publishing, λ_p, is higher when the size of the network, n, increases. This is because the utility of the attacker decreases in n^α. This is similar to the pure strategy equilibrium: when n is large, "not attacking" is a dominant strategy for the attacker. In this case, the best response for the publisher is to publish, as long as $V_p \geqslant 1$. The same reasoning can be applied to the cost of attacking: the higher it is, the higher the probability of publishing. When $V_a = n^\alpha$, the publisher always publishes ($\lambda_a = 1$), as not attacking is a dominant strategy for the attacker.

Having found λ_a and λ_p, we can calculate the expected utility of each of the players at this equilibrium:

$$\begin{aligned}EU_a &= V_a - n^\alpha \\ EU_p &= 0\end{aligned}$$

These payoffs are exactly the same as in the pure strategy case when the attacker attacks and the publisher doesn't publish. This pure strategy case, however, is not an equilibrium: knowing that the publisher doesn't publish, the attacker would prefer not to attack, but knowing that, the publisher would then be willing to publish. Thus the two players must instead play their two pure strategies with probabilities λ_p and λ_a.

5 Multiple publishers

Thus far we have only considered a network where multiple copies of a single document are disseminated by one publisher. Or, from another perspective, the attacker gains no

utility from censoring other documents as a consequence of attacking a particular target document. This is quite appropriate for the case of censorship by the Church of Scientology, which might be interested in removing their secrets from the network but do not care about the Communist Manifesto. On the other hand, there are censorship authorities who might be interested in removing more than one document from the network.

In this model, we have p publishers, and each distributes d copies of her own document. They are participating in a network of n nodes, of which the attacker has chosen to corrupt a proportion, x, of nodes.

The new expected utility function of the attacker is:

$$EU_a = px^d - nx \tag{22}$$

This is merely the expected value of a binomial distribution across the number of publishers whose documents are only stored on corrupt nodes. For an individual publisher, the probability that his document is censored is x^d, and there are p publishers in total. The intuition behind this is that the attacker's utility increases linearly with the number of documents that are no longer accessible.

The attacker's utility increases with the number of different documents he can suppress with the same effort. This is particularly true when there is only a small number of copies of each document distributed to nodes in the network. On the other hand, increasing the size of the network will decrease the attacker's utility.

6 Analysis of censorship-resistance properties

At this point it is useful to stand back from the results presented in the previous sections and look at what they might mean in the context of a real censorship-resistance system. As stated in the introduction, our threat model is rather different from that specified in, for example, [Dou02], [SM02] or [CDG+02], because we consider not only whether it is possible for an attacker with given resources to compromise the system, but also whether a rational attacker would devote his resources given the expected payoff.

First of all, rather trivially, we have confirmed that the security of the censorship resistant network depends on two factors – the size of the network and the value of the document(s) it is storing.

Size of the network Increasing the size of the network is bound to decrease the utility of the attacker (holding everything else constant), as there is more chance of his attacks 'missing' nodes containing documents. It will also increase the utility of the publisher, as there is more chance of his documents surviving. If it is possible for the publisher to increase the size of the network, this is a good strategy for defending his documents against censorship.

Value of the documents The attacker attaches some relative utility to successfully censoring a document against the cost of attack; V_a. There are certainly situations where it would never be worthwhile to attack, because the cost of attack is so high, or the utility attached to censorship is low. An example of this would be a network containing only the holiday photos of its users: it is hard to imagine an attacker that would attach great utility to preventing the distribution of these 'documents'. V_a might be much higher if, for example, the documents contain libellous statements about the attacker. It can also be observed at this point that merely introducing more distinct documents into the network (in the multiple publisher case) can increase the utility of an indiscriminate attacker (who does not care which particular document he censors). This, in turn, increases the likelihood of an attack.

Our new threat model depends heavily on the cost of compromising the nodes to the atacker. We investigated both linear and polynomial cost in the number of nodes compromised (it is impossible to tell which one is more realistic).

If the cost of compromising nodes is linear in the number of nodes, our model showed that either the entire network is compromised, or none of it is. Note, of course, that the traditional threat models predict that the network will always be compromised (contrary to the example of holiday photo sharing network).

If the cost is polynomial in the number of nodes, we see more interesting behaviour: it makes perfect sense (i.e. there is an equilibrium) for the attacker to compromise a fraction of nodes and for the publisher to publish to several nodes. This is precisely the situation that traditional models from Computer Science miss[1].

7 Sequential search

In previous sections we assumed that a document could be retrieved if at least one copy was published to a node that was not corrupted by the attacker. We now extend our model by introducing another agent: the retriever. Her aim is to retrieve a single copy of the document by searching nodes at random, but without replacement (i.e. she will never search the same node twice). She has a cost associated with searching a single node, and after each search decides whether to continue the search, or to give up. There is also a utility associated with finding a copy of the document, U_r. The optimal decision rule for the retriever is to continue searching if and only if the probability of finding the document multiplied by the utility associated with successful retrieval is greater than the expected cost of continuing the search.

We first find the probability that the search will terminate after a given number of steps. Given that the search stops when the first copy of the document is found, the first observation to make is that if the search has continued for n steps, on the previous $n-1$ steps the nodes searched must have either

[1]However, this only occurs provided the number of nodes the publisher chooses to publish to is small (smaller than the parameter α of the cost function).

- received a copy of the document but been corrupted, or
- not received a copy of the document but been corrupted, or
- not received a copy of the document and not been corrupted.

Considering these cases in order, we first find the probability that a node picked at random from the n remaining to be searched (of which p have received copies of the document and b have been corrupted) has received a copy of the document and has been corrupted. We denote this probability $\beta(n, p, b)$. We alter one of our assumptions at this point to make the analysis simpler: both the corrupt nodes and the nodes published to are now selected without replacement. This avoids the need to consider cases where a node has received a copy of a document more than once.

$$\beta(n, p, b) = \frac{bp}{n^2} \tag{23}$$

The probabilty that a node has not received a copy of the document but has been corrupted is denoted $\gamma(n, p, b)$. The function is defined as

$$\gamma(n, p, b) = \left(1 - \frac{p}{n}\right) \frac{b}{n} \tag{24}$$

$\delta(n, p, b)$ is the probability that a node has neither been corrupted nor received a copy of the document.

$$\delta(n, p, b) = \left(1 - \frac{p}{n}\right) \left(1 - \frac{b}{n}\right) \tag{25}$$

Finally, we need to consider the probability with which we terminate our search, i.e. the probability that we find a node that both received a copy of the document, and was not corrupted. This we denote $\alpha(n, b, p)$.

$$\alpha(n, b, p) = \frac{p}{n} \left(1 - \frac{b}{n}\right) \tag{26}$$

We abuse our notation slightly to call the events that occur with these probabilities α, β, γ and δ also. The possible outcomes of a search can thus be described using sequences of these letters. For example, $\beta\gamma\alpha$ means that

- the first node searched both received a copy of the document and was corrupted;
- the second node searched was corrupted, but never received a copy of the document; and
- the third node searched received a copy of the document, and was not corrupted.

All possible searches can thus be represented by the language of strings made up from the characters α, β, γ and δ, subject to the constraints that

- they are of a most length N, where N is the number of nodes in the network, and
- they contain at most one α, which must occur at the end of the string

We now denote the strings that represent the possible searches before termination (i.e. not containing an α) as ρ_i, where i is the length of the search. The null search is ρ_0. We take $\beta\rho_{i-1}$ as a shorthand for 'the set made up of the concatenation of β with each string in ρ_{i-1}'.

$$\rho_0 = \{\} \tag{27}$$
$$\rho_1 = \{\beta, \gamma, \delta\} \tag{28}$$
$$\rho_2 = \{\beta\beta, \beta\gamma, \beta\delta, \gamma\beta, \gamma\gamma, \gamma\delta, \delta\beta, \delta\gamma, \delta\delta\} \tag{29}$$
$$\rho_i = \{\beta\rho_{i-1}, \gamma\rho_{i-1}, \delta\rho_{i-1}\} \tag{30}$$

Allowing these sets to be indexed further, we take $\rho_{i,1}$ to be the first possible search scenario of length i (all events were β), and $\rho_{i,3^i}$ to be the last (all events were δ). It is important to know that each time a new search takes place (with outcome β, γ or δ), we must update the values of n, p and b accordingly. If an event of type β occurs, n, p, and b should all be decreased by one (i.e. there is one fewer node in the system to be searched, and it was both a node that was published to, and a node that was corrupted). For an event of type γ, n and b should be decreased by one, and p should remain as it is. Finally, if an event of type δ occurs, only n should be decreased by one.

The final tool that we need is a function to count the number of occurrences of β, γ and δ in a particular sequence of events described by $\rho_{i,j}$. Taking, for example, $\rho_{3,2}$ (which evaluates to '$\beta\beta\gamma$'), $\mathbf{count}(\beta, \rho_{3,2})$ evaluates to 2. This allows us to work out at any stage of the search how many times n, p and b should have been decremented.

We are now in a position where we can evaluate the probability of a search terminating after any number of steps (up to the number of nodes in the network), by simply summing the probabilities of the events described by $\rho_{i,j}$, and adding α events for termination as necessary. We can furthermore work out the probability that the document will never be found, even if all nodes in the network are searched.

To make this analysis clearer, we present a concrete example of the search problem. We set the size of the network, $N = 3$, the number of nodes to which the document was sent, $P = 2$, and the number of nodes which were attacked, $B = 2$. The strings describing the possible outcomes (with non-zero probability) where a successful retrieval occurs are:

Search sequence	Expands to	Probability
α	α	2/9
$\rho_{1,1}\alpha$	$\beta\alpha$	1/9
$\rho_{1,2}\alpha$	$\gamma\alpha$	1/9
$\rho_{2,2}\alpha$	$\beta\gamma\alpha$	1/9
$\rho_{2,4}\alpha$	$\gamma\beta\alpha$	1/9
$\rho_{3,3}\alpha$	$\beta\beta\delta$	1/9
$\rho_{3,7}\alpha$	$\beta\delta\beta$	1/9
$\rho_{3,19}\alpha$	$\delta\beta\beta$	1/9

As an example, we now run through the derivation of the probability for the sequence $\gamma\beta\alpha$. Taking the updating rules into account, we obtain

$$\begin{aligned} P(\rho_{2,4}\alpha) &= P(\gamma\beta\alpha) \\ &= \gamma(N,P,B) \times \beta(N-1,P,B-1) \times \alpha(N-2,P-1,B-2) \\ &= \left(1-\frac{P}{N}\right) \times \frac{B}{N} \times \frac{(B-1)P}{(N-1)^2} \times \frac{P-1}{N-2} \times \left(1-\frac{B-2}{N-2}\right) \\ &= 1/9 \end{aligned}$$

These probabilities are independent, and can thus be summed to make more general statements about the length of the search. For example, the probability that the search finishes after one step is 2/9, the probability that it finishes after two steps is 2/9, and the probability it finishes successfully after three steps is 2/9. The probability that the document is never found is 1/3. With these tools, it is thus possible for a retriever to make an optimal decision at any point in the search, as he knows both the expected cost of continuing the search and the probability of success in the search. Furthermore, we can assess in advance the maximum length of search that a retrieving agent will be willing to make for given values of N, P, B, U_r (utility attached to successful retrieval) and c_r (cost of searching a given node).

8 Further work

In order to make our models more applicable to currently deployed systems, a few changes will have to be made. It is unrealistic to model publishing with the 'entire document to d nodes' paradigm; it is far more likely that the document would be deployed as shares, such that any k shares would allow reconstruction of the entire document. Our model of corruption does not take the entire picture into account either: the attacker may well have scope for adding his own malicious nodes to the network, rather than just corrupting existing ones.

Further development of our search model may also be possible. In some systems it is quite possible that although a document exists on some node (or nodes), the cost of searching would make its retrieval unviable. The sequential search goes some way towards this ideal, but there is still work to be done, for example with different cost functions.

In our basic model we have assumed that however many nodes the attacker corrupts, the network stays completely connected. This may not be the case. We may wish to take into account the topology of the network in order to assess its vulnerability.

It would also be interesting to apply our model to other scenarios such as distributed computation. For instance, one might imagine computations which consists of three independent parts, all of which need to succeed, with the defender being able to launch several copies of each.

9 Conclusions

In this paper we argue that an economic perspective on the security of peer-to-peer systems can supplement existing models (e.g. that proposed by Doceur). In particular, we propose that participants in a peer-to-peer network are economically rational: they seek to maximize their utility.

In our basic model, in which the marginal cost of compromising each extra node is constant, the attacker is best served either by compromising the entire network or none of it. Perhaps unsurprisingly, this is not the result when the marginal cost of compromise is more realistically modelled as increasing in the number of nodes compromised.

When we model non-linear cost of attack, we find a large class of solutions where the attacker still attacks the entire network or none of it (the case where $d > \alpha$). We are, however, now able to find pure strategy Nash equilibria, where the attacker attacks only some of the network, and the publisher publishes to some (for realistic values of α) small number of nodes.

In the multiple publisher case, we find that the attacker is more likely to attack when, by doing so, he can suppress more documents; he accomplishes more with the same amount of work. However, the attacker's expected utility decreases when publishers insert more copies of each document into the network, because there is a greater chance of his attack 'missing' a copy.

These two models lead to a clear strategy by which a publisher can increase the security of the network: increase the number of copies on the network, increase the size of the network, or increase the cost to the attacker of corrupting nodes.

Our model of sequential search has also proved fruitful, in that we now have an algorithm describing the optimal decision rule for an agent retrieving documents in the system. For a given set of system parameters, and given a search sequence that has not yet yielded the document, we can find the probability that the document will be retrieved (and the expected cost of this retrieval). We can also find the probability that no node will have

a copy of the document (i.e. that the search will never succeed).

Designing totally secure peer-to-peer networks may be impossible. Fortunately, when the problem is reconsidered as designing *sufficiently secure* peer-to-peer networks, it becomes much more tractable. We believe that considering peer-to-peer networks from an economic perspective provides valuable insight into the means of creating practical, usable, sufficiently secure peer-to-peer networks.

References

[ACK+02] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: an experiment in public-resource computing. *Commun. ACM*, 45(11):56–61, 2002.

[And96] R. J. Anderson. The Eternity Service. In *Pragocrypt*, 1996. `http://www.cl.cam.ac.uk/users/rja14/eternity/eternity.html`.

[CDG+02] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *5th Usenix Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.

[CSWH01] Ian Clarke, Oscar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In Federrath [Fed01], pages 46–66. `http://freenet.sourceforge.net`.

[DFM01] Roger Dingledine, Michael J. Freedman, and David Molnar. The Free Haven Project: Distributed anonymous storage service. In Federrath [Fed01], pages 67–95. `http://freehaven.net`.

[Dou02] John R. Douceur. The Sybil Attack. In *First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, Cambridge, MA, March 2002.

[Fed01] Hannes Federrath, editor. *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.

[GLBML01] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, and Mark Lillibridge. Incentives for sharing in peer-to-peer networks. *Lecture Notes in Computer Science*, 2232, 2001.

[HR02] Steven Hand and Timothy Roscoe. Mnemosyne: Peer-to-Peer Steganographic Storage. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), Boston, MA, USA*, March 2002.

[Ser02] Andrei Serjantov. Anonymizing censorship resistant systems. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, March 2002.

[SM02] Emil Sit and Robert T. Morris. Security considerations for peer-to-peer distributed hash tables. In *First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, Cambridge, MA, March 2002.

[WRC00] Marc Waldman, Aviel D. Rubin, and Lorrie Faith Cranor. Publius: A robust, tamper-evident, censorship-resistant, web publishing system. In *Proc. 9th USENIX Security Symposium*, August 2000.