How to Spread Adversarial Nodes? Rotate!

Christian Scheideler[*]
Dept. of Computer Science
Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218, USA
scheideler@cs.jhu.edu

ABSTRACT

In this paper we study the problem of how to keep a dynamic system of nodes well-mixed even under adversarial behavior. This problem is very important in the context of distributed systems.

More specifically, we consider the following game: There are n white pebbles and ϵn black pebbles for some fixed constant $\epsilon < 1$. Initially, all of the white pebbles are laid down in a ring, and the adversary has all of the black pebbles in its bag. In each round, the adversary can look at the entire ring and can select to add a black pebble to the ring (if its bag is not empty) or to take any black pebble from the ring and put it back into its bag (i.e. we consider *adaptive* adversaries). However, the adversary cannot place a black pebble into any position it likes. This is handled by a join strategy to be specified by the system. The goal is to find an *oblivious* join strategy, i.e. a strategy that cannot distinguish between the white and black pebbles in the ring, that integrates the black pebbles into this ring and may do some further rearrangements so that for a polynomial number of rounds the adversary will not manage to include its black pebbles into the ring so that there is a sequence of $s = \Theta(\log n)$ consecutive pebbles in which at least half of the pebbles are black. If this is achieved by the join strategy, it wins. Otherwise, the adversary wins.

Of course, the brute-force strategy of rearranging all of the pebbles in the ring at random after each insertion of a black pebble will achieve the stated goal, with high probability, but this would be a very expensive strategy. The challenge is to find a join strategy that needs as little randomness and as few rearrangements as possible in order to win with high probability. In this paper, we present and analyze a very simple strategy called *k-rotation* that chooses $k-1$ existing positions uniformly at random in the ring, creates a new position uniformly at random in the ring, and then rotates the new pebble and the $k-1$ old pebbles along these positions. Interestingly, even if the adversary has just s pebbles, it can still win for $k = 2$. But the k-rotation rule wins with high probability for $k = 3$ as long as $\epsilon < 1/3$, demonstrating that there is a sharp threshold for keeping pebbles in a sufficiently perturbed state.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Routing and layout*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms and network problems*

General Terms

Algorithms, Theory

Keywords

Random mixing, proactive security, join-leave attacks

1. INTRODUCTION

In this paper we study the problem of how to keep a dynamic system of nodes well-mixed even under adversarial behavior. This problem is very important in the context of distributed systems.

Research on scalable distributed systems has recently received an enormous attention due to the popularity of peer-to-peer systems. Many scalable designs for peer-to-peer systems are already known, such as Chord [33], CAN [29], Pastry [16], and Tapestry [34], to name just a few. However, nowadays, scalability is not the only concern but robustness against adversarial behavior becomes an increasingly pressing issue. The *robustness* of a distributed system is measured in its ability to withstand massive and malicious attacks, including Byzantine behavior of its components. Achieving robustness and scalability at the same time is a decades old open problem in the field of distributed computing. The biggest threats appear to be insider attacks and distributed denial-of-service (DoS) attacks, against which much of the security measures are ineffective. These threats are particularly hard to avoid in *open* distributed systems, i.e. systems where mutually distrusting parties with conflicting interests are allowed to join (such as the Internet).

There are basically two kinds of approaches of making a distributed system robust against adversarial behavior: *proactive* and *reactive* approaches. Proactive approaches aim at *protecting* a system against attacks, thereby achieving a high availability of the distributed system, whereas reactive approaches are based on mechanisms to *detect* and *recover* from attacks. Proactive strategies are known to be expensive and cannot protect against all possible bad events that may happen. On the other hand, just using reactive strategies may sacrifice the availability of a distributed system in critical situations. Since it is desirable to have a highly available system, the proactive security measures should therefore be pushed as far as possible without paying too much overhead. But how far can this be pushed, and can any formal guarantees be given?

[*]Supported by NSF grant CCR-0311121 and NSF grant CCR-0311795.

STOC'05, May 22–24, 2005, Baltimore, Maryland.
Copyright 2005 ACM 1-58113-960-8/05/0005 ...$5.00.

A prerequisite for a distributed system to work correctly is that an overlay network can be maintained between its sites. Once distributed systems become large enough, one *has* to deal with sites continuously entering and leaving the system, simply because sites may fail and have to be replaced by new sites or because additional resources have to be added to preserve the functionality of the system. Hence, in general, a distributed system supporting any service has to have an overlay network supporting joining, leaving and routing between the sites, and without a robust and scalable implementation of such a network, the field of scalable and robust distributed services does not really exist.

Most research on robust distributed systems in the past has ignored the issue that sites may continuously join and leave the system. Proactive security has mostly been studied in the context of a fixed, distributed system. Proactively secure solutions have been devised, for example, for secret sharing [22, 27], public key signatures [5, 7, 17, 18, 20, 23, 28], link security and secure end-to-end communication [8], and (pseudo-)random number generation [9, 12]. These solutions can be extended to dynamic distributed systems if one is willing to pay a linear overhead, i.e., any object that needs proactive protection is managed by *all* of the sites. Alternative approaches are known that just need $O(\sqrt{n})$ of the n sites to manage any object in order to protect it in a dynamic environment [1, 24, 25], but for an approach to be truly scalable, only $O(\log n)$ of the n sites should be involved in managing any object.

Peer-to-peer systems such as Chord [33] only require an object to be stored among $O(\log n)$ sites and are therefore scalable. However, the robustness of these systems hinges on the assumption that the unreliable and/or adversarial sites are *randomly* distributed in the system so that for every object, only a minority of the sites responsible for the object may create problems, with high probability. But this randomness assumption is problematic if sites are allowed to join and leave the system, especially in open distributed systems.

Typically, when sites join a peer-to-peer system, they are either given a random ID or an ID based on a one-way hash value of their IP address. Both approaches open up the possibility for an adversary to degrade randomness by using *join-leave attacks*. In the latter case, it just needs sufficiently many IP addresses (which will not be a problem with IPv6), and in the former case, it just needs to repeatedly join and leave the system with its sites until it is able to place its sites into certain areas of the system. People in the systems community are well aware of the danger of these and similar attacks [15, 13], and various solutions have been proposed that may help thwart these attacks in practice [10, 26, 31, 32], but until recently no mechanism was known that can *provably* cope with these attacks without sacrificing the openness of the system.

The only mechanism known so far that can preserve randomness in the system uses *random* node IDs and enforces a *limited* lifetime on every node in the system, i.e., every node *has* to reinject itself after a certain amount of time steps [3]. But this leaves the system in a hyperactive mode that may unnecessarily consume resources that could be better used for other purposes. Ideally, one would like to use *competitive* strategies. That is, the resources consumed by the proactive security mechanism should scale with the join-leave activity of the system. This will make sure that if there is no join-leave activity, the proactive mechanism will be idle as well. Can we design a competitive strategy with low overhead? We will show that this is possible.

1.1 How to counter join-leave attacks

It has been shown in [3] that no open, predictable overlay network can survive adaptive adversarial join-leave attacks. Hence, as a prerequisite, we have to use randomness. Our basic approach is to assign to each node a random place in the system. However, as mentioned above, we also have to do some further rearrangements in addition to this to prevent adaptive attacks on the system.

If we want to have a competitive strategy, rearrangements of nodes in the system should only be triggered by join or leave requests. Leave requests are problematic because the adversary cannot be forced to initiate any rearrangement strategy in a correct way when it leaves. However, for join requests, rearrangements *can* be enforced because the adversary *wants* to get into the system. Verifiable secret sharing approaches may be used here (e.g., [4, 6, 11, 19, 21]) because they can be used to generate unbiased random numbers even if adversarial nodes are involved in it. Verifiable secret sharing consists of a sharing phase and a recovery phase. The sharing phase allows honest nodes to obtain a proof whether the secret (e.g., the random numbers for the join request) is recoverable without revealing the secret. Once an honest node obtains such a proof, this can be used to *enforce* the correct execution of the join operation by presenting the proof to the other participants. Without such a proof an honest node will not participate in the recovery phase.

Hence, the only way the adversary can avoid rearrangements in the join operation is to prevent an honest node from getting a proof, but then it cannot join and, crucially, also does not learn about the random numbers for the join operation.

Now, how many rearrangements should be made for each join request? Here, one may recall a well-known fact from card shuffling. Consider the situation that a deck of n cards is laid out in a row. In a random transposition operation, we pick two random cards and exchange their positions. The question is, how many operations are necessary to achieve a random permutation of the cards. Diaconis and Shahshahani [14] showed that $O(n \log n)$ random transpositions are sufficient for this, and it is also known that this bound is tight. Hence, it seems appropriate in our setting to perform $\Theta(\log n)$ random transpositions for each join request to keep the system in a random state. But it turns out one can do much better than that. We investigate this by focusing on a specific game.

1.2 The game

Consider the following game: There are n white pebbles and ϵn black pebbles for some fixed constant $\epsilon < 1$. Initially, all of the white pebbles are laid down in a ring, and the adversary has all of the black pebbles in its bag. In each round, the adversary can look at the entire ring and can select to add a black pebble to the ring (if its bag is not empty) or to take any black pebble from the ring and put it back into its bag (i.e. we consider *adaptive* adversaries). However, the adversary cannot place a black pebble into any position it likes. This is handled by a join strategy to be specified by the system. The goal is to find an *oblivious* join strategy, i.e. a strategy that cannot distinguish between the white and black pebbles in the ring, that integrates the black pebbles into this ring and may do some further rearrangements so that for a polynomial number of rounds the adversary will not manage to include its black pebbles into the ring so that there is a sequence of $s = \Theta(\log n)$ consecutive pebbles in which at least half of the pebbles are black. If this is achieved by the join strategy, it wins. Otherwise, the adversary wins.

To the best of our knowledge, the game has not been studied before. Therefore, let us motivate why we believe that it is relevant for distributed systems. First of all, one may ask why only the black pebbles join and leave the system. The reason is that this actually represents the worst case for a dynamic distributed system. All schemes we are aware of for perturbing the pebbles actually work better the more white pebbles join and leave the system. Second, the fact that white pebbles are never turned into black pebbles appears to be a limitation. However, our idea behind the black and white pebbles is that black pebbles are owned by the adversary whereas white pebbles are owned by honest peers. White pebbles may still have vulnerabilities that the adversary may try to exploit, but whether or not a peer is vulnerable is usually not under the control of the adversary, so that it suffices to assume an initially random layout of the white pebbles in order to prevent the adversary from winning on the white pebbles. Finally, the game does not look at concurrency. However, it follows from random graph theory that as long as at most δn requests are served simultaneously at any time for some sufficiently small constant $\delta > 0$, concurrent executions of join strategies similar to our proposed strategy do not form cycles up to a negligible fraction, with high probability, so that up to a negligible fraction, the concurrent executions are serializable and we therefore expect the outcome to be almost the same as their sequential execution. Thus, for simplicity, we leave the concurrency aspect out of this paper.

1.3 Why the game?

Still, one may ask what is so important about having a majority of white pebbles in any sequence of s pebbles on the ring? Recall that an overlay network needs to support joining, leaving, and routing as basic primitives. Consider the ring studied in our game, and interpret every pebble as a node. Suppose that a proper join operation can be designed so that for any sequence of s consecutive nodes, the number of white (i.e., well-behaved) nodes is in the majority. Using this property, we can design a robust routing strategy for the ring network. We call a routing strategy *robust* if every message sent out by a well-behaved node is guaranteed to reach its destination. This is more difficult than it seems because the adversarial nodes may not just try to delete or alter the message but also to generate many messages by themselves to prevent some message from ever reaching its destination. The following approach can help here:

Suppose that each node maintains connections to its s closest successors and its s closest predecessors on the ring. If a node v wants to route a message M to node w, it sends M to all of its successors. Each neighbor of w checks whether w has already exceeded its allowed rate of message injections. If so, it rejects M and otherwise accepts M. Every node accepting M forwards it to all of its known successors. Every node that receives a message M from at least half of its predecessors forwards it to all of its successors (or the destination, if it is known).

It is easy to show that if the well-behaved nodes are in the majority in every sequence of s consecutive nodes, this routing strategy is indeed robust. Certainly, scalability is still an issue, which may be handled using approaches similar to skip graphs [2], but this is beyond the scope of this paper. The goal of this paper is solely to demonstrate that it is possible to counter adaptive join-leave attacks using efficient, competitive mixing strategies.

1.4 Our contribution

We propose the *k-rotation* strategy in order to randomly perturb the pebbles. The k-rotation strategy works as follows: Initially, the new black pebble is declared a homeless pebble. For $k-1$ rounds, place the currently homeless pebble into a random position of the ring and declare the pebble previously placed at that position the new homeless pebble. Afterwards, create a new position at a random place in the ring and place the homeless pebble there.

It turns out that $k \leq 2$ is not sufficient but $k \geq 3$ is sufficient for the system to win with high probability. Interestingly, the adversary has a good chance of winning for $k = 2$ even if it has only $O(\log n)$ pebbles, whereas the adversary has only a negligible chance of winning for $k = 3$, even when having $n/4$ pebbles. Thus, a sharp threshold can be identified for the system to win or lose. Our results are summarized in the following theorem.

THEOREM 1.1. *Let n and $s = O(\log n)$ be sufficiently large. When using the k-rotation strategy, it holds:*

- *If $k = 1$, then the adversary only needs $s/2$ pebbles to win within $O(n)$ join attempts, with high probability.*
- *If $k = 2$, then the adversary only needs s pebbles to win within $O(n \log s)$ join attempts on expectation and within $O((n \log s) \log n)$ join attempts, with high probability.*
- *If $k \geq 3$, then the adversary loses with high probability as long as it has $\leq \epsilon n$ nodes for some constant $\epsilon < 1 - 2/k$, and this result is tight.*

In fact, the k-rotation rule ensures that for any $k \geq 3$, the fraction of black pebbles in a sequence of s consecutive pebbles is at most

$$(1+\delta)\left(\frac{1+k\epsilon}{k+k\epsilon}\right)$$

with high probability, where $\delta > 0$ can be an arbitrarily small constant depending on s.

Thus, as k increases, ϵ can get arbitrarily close to 1. Note that ϵ must be smaller than 1 because otherwise there is certainly no chance for the system to win.

2. ANALYSIS OF THE K-ROTATION RULE

Recall the k-rotation strategy from the previous section. We analyze the worst-case scenario separately for $k = 1$, $k = 2$, $k = 3$, and all even $k > 3$.

2.1 Outcome for $k = 1$

For $k = 1$, all what happens in a join operation is that a new position is created uniformly at random in the ring and the new, black pebble is placed into it. This scenario makes it very easy for the adversary to win:

Focus on some fixed sequence S of $s/2$ consecutive positions on the initial ring. Continue to inject the black pebbles into the system until they are all inside of S. Each black pebble that does not land inside of S is taken out again and reinjected into the system.

Using this strategy, the following result can be shown, which implies that the adversary can quickly gain the majority of pebbles in a sequence of size s.

LEMMA 2.1. *Consider any sequence S of $s/2$ consecutive positions on the initial ring. If the adversary has at least $s/2$ black pebbles, then it takes at most $O(n)$ join requests until the adversary has at least $s/2$ black pebbles inside of S, with high probability.*

PROOF. For any $i \geq 1$, let the binary random variable X_i be 1 if and only if in the ith join request the adversary manages to get a black pebble into S. For each join request, the probability is at least $(s/2-1)/n$ that the black pebble is placed inside of S. Hence, $\Pr[X_i = 1] \geq (s/2-1)/n$. Consider now the random variable $S_t = \sum_{i=1}^{t} X_i$. Certainly, $\mathrm{E}[S_t] = \sum_{i=1}^{t} \mathrm{E}[X_i] \geq t\cdot(s/2-1)/n$. Furthermore, because the X_i's are independent, it follows from the Chernoff bounds (e.g., [30]) that for any $\epsilon \in [0,1]$,

$$\Pr[S_t \leq (1-\epsilon)\mathrm{E}[S_t]] \leq e^{-\epsilon^2 \mathrm{E}[S_t]/2} .$$

For $t \geq n/(1-\epsilon) \cdot 1/(1-2/s)$ it holds that $\Pr[S_t \leq s/2] \leq e^{-\epsilon^2 n/3(1-\epsilon)}$. This is exponentially small in n for any constant $\epsilon \in (0,1)$. Because $t = O(n)$ in this case, the theorem follows. □

2.2 Outcome for $k = 2$

In order to analyze this case, we associate a sequence of s consecutive pebbles with every initial position in the system. Given a position p, we define S_p as the sequence consisting of the s closest successors of p on the ring (in clock-wise direction). This property is maintained for S_p as new positions are created and old positions are removed. If p is removed itself, the predecessor of p on the ring takes over p's role. Thus, S_p is well-defined at any time.

Consider now any fixed sequence S_p for some initial position p in the ring. For simplicity, we just call it S in the following, and we assume the pebbles in S to be laid out from left to right with p being to the left of S. Consider an adversarial strategy that never issues a leave request for a black pebble in S (but may decide to remove any black pebble outside of S at any time). In this case, we only have to focus on the effect of join requests on S.

In order to analyze the effect of join requests, we split the execution of a join request into several stages. In stage 0 the adversary just presents the new black pebble to the system, which we declare as *homeless*. Stage $i \in \{1, \ldots, k-1\}$ represents the ith replacement of a pebble, which places the currently homeless pebble into the ith position and declares the pebble formerly placed in this position as the new homeless pebble. Stage k represents the point at which a new position is created in the ring and the currently homeless pebble is placed into it.

We model the effect of these stages as a stochastic process. Let $U = \{(i,j) \mid i \in \{0, \ldots, s\},\ j \in \{0,1\}\}$ be the state space of sequence S. In state (i,j), i represents the number of black pebbles in S, and $j = 1$ if and only if the currently homeless pebble is black. Let $P_t = (p^{(t)}_{u,v})_{u,v \in U}$ represent the system of transition probabilities at stage t, i.e., $p^{(t)}_{u,v}$ represents the probability of moving from state u to state v in stage t. We will determine P_t for the various stages.

Stage 0: Because the only event that happens is that the new black pebble is declared homeless, we have

$$p^{(0)}_{(i,0),(i,1)} = 1 \quad \text{and} \quad p^{(0)}_{(i,1),(i,1)} = 1 \quad \text{for all } i \in \{0, \ldots, s\}$$

and all other transition probabilities are 0.

Stage t for some $t \in \{1, \ldots, k-1\}$: Let N denote the total number of positions in the ring at the beginning of the join operation and n denote the number of white pebbles. If S is currently in the state $(i,0)$, then S stays at $(i,0)$ if a position with a white pebble is selected, S changes to $(i-1,1)$ if the homeless pebble is placed in a position with a black pebble in S, and S changes to $(i,1)$ if the homeless pebble is placed in a position with a black pebble outside of S. Hence, for all $i \in \{0, \ldots, s\}$,

$$p^{(t)}_{(i,0),v} = \begin{cases} i/N & : \quad v = (i-1,1) \\ n/N & : \quad v = (i,0) \\ (N-n-i)/N & : \quad v = (i,1) \end{cases}$$

If S is currently in the state $(i,1)$, then S stays at $(i,1)$ if a position with a black pebble is selected, S changes to $(i,0)$ if the homeless pebble is placed in a position with a white pebble outside of S, and S changes to $(i+1,0)$ if the homeless pebble is placed in a position with a white pebble in S. Hence, for all $i \in \{0, \ldots, s\}$,

$$p^{(t)}_{(i,1),v} = \begin{cases} (n-s+i)/N & : \quad v = (i,0) \\ (N-n)/N & : \quad v = (i,1) \\ (s-i)/N & : \quad v = (i+1,0) \end{cases}$$

In all other cases, the transition probabilities are equal to 0.

Stage k: Suppose that S is currently in state $(i,0)$. Then S changes to $(i-1,0)$ if the homeless pebble is placed inside or directly to the left of S and a position with a black pebble gets evicted from S. In all other cases, S stays in the state $(i,0)$. In order to determine the probability that a black pebble gets evicted, we need the following lemma.

LEMMA 2.2. *Given that the adversary never removes a pebble from S, every pebble in S is equally likely to get evicted from S.*

PROOF. Let the initial pebbles in S be numbered from 1 to s and the jth pebble that joins S (either due to a replacement or the creation of a new position) be given number $s+j$. Let Π_j be the set of all possible permutations of the pebbles in S after the jth event that a pebble joins S. We prove by induction that for every $j \geq 0$, every $\pi \in \Pi_j$ is equally likely to occur. This will immediately imply the lemma.

Under the assumption that the white pebbles are initially randomly distributed on the ring, the induction hypothesis is certainly true for $j = 0$. So suppose that it is true up to some j. In order to show that it is also true for $j+1$, we consider two cases:

Suppose that at some stage $i < k$, the homeless pebble p takes over the position of some pebble q in S. Since p chooses each position in S with the same probability, it follows from the hypothesis that every permutation of the pebbles in S remains to be equally likely, no matter which q is chosen.

Suppose that at stage k, the homeless pebble is placed into a new position in S. For any pebble q in S, let $\Pi_{j,q}$ be the set of all permutations with q on the right side. Recall that every permutation is equally likely to occur. Hence, when removing q from S we obtain a set $\Pi'_{j,q}$ of all permutations of the remaining pebbles that are equally likely to occur. Since the new pebble may be added at any of the s possible places in these permutations, and every case is equally likely to occur, we end up with a set Π_{j+1} of all permutations in which each is equally like to occur. □

It follows from the lemma that for the adversarial strategy considered by us, the probability of a particular pebble to get evicted is independent of the history of S but only depends on the current number of black pebbles in S. More precisely, the probability that a black pebble is evicted from S is i/s. Therefore, for all $i \in \{0, \ldots, s\}$,

$$p^{(k)}_{(i,0),v} = \begin{cases} \frac{i}{s} \cdot \frac{s}{N} = \frac{i}{N} & : \quad v = (i-1,0) \\ 1 - \frac{i}{s} \cdot \frac{s}{N} = 1 - \frac{i}{N} & : \quad v = (i,0) \end{cases}$$

If S is currently in state $(i,1)$, then S changes to $(i+1,0)$ if the homeless pebble is placed inside or directly to the left of S and a position with a white pebble gets evicted from S. In all other cases, S changes to the state $(i,0)$. From the lemma above it follows that the probability that a white pebble is evicted from S is $1 - i/s$. Hence, for all $i \in \{0, \ldots, s\}$,

$$p^{(k)}_{(i,1),v} = \begin{cases} 1 - \left(1 - \frac{i}{s}\right) \cdot \frac{s}{N} = 1 - \frac{s-i}{N} & : \quad v = (i,0) \\ \left(1 - \frac{i}{s}\right) \cdot \frac{s}{N} = \frac{s-i}{N} & : \quad v = (i+1,0) \end{cases}$$

Suppose now that $k = 2$ and that S is in the state $(i,0)$ before executing the k-rotation. Then it holds

- after stage 0: S is in the state $(i,1)$
- after stage 1: S has a probability distribution of

$$(q_{(i,0)}, q_{(i,1)}, q_{(i+1,0)}) = \left(\frac{n-s+i}{N},\ \frac{N-n}{N},\ \frac{s-i}{N}\right)$$

- after stage 2: S has a probability distribution of $(q_{(i-1,0)}, q_{(i,0)}, q_{(i+1,0)})$ with $q_{(i-1,0)} = (i/N) \cdot (n-s+i)/N$ and
$$q_{(i+1,0)} = \frac{s-i}{N}\left(\frac{N-n}{N} + \left(1 - \frac{i+1}{N}\right)\right)$$

Thus, we can model the effect of a join operation on S as a simple birth-death process P on the state space $U' = \{0, \dots, s\}$ with transition probabilities

$$p_{i,j} = \begin{cases} \frac{i(n-s+i)}{N^2} & : \quad j = i-1 \\ \frac{s-i}{N}\left(2 - \frac{n+i+1}{N}\right) & : \quad j = i+1 \\ 1 - (p_{i,i-1} + p_{i,i+1}) & : \quad j = i \end{cases} \qquad (1)$$

In order to determine the behavior of this birth-death process, we need a series of lemmas. The first is well-known, but we sketch its proof for completeness.

LEMMA 2.3. *Any birth-death process P on a state space $U = \{0, \dots, s\}$ with transition probabilities $p_{i,i+1} = \lambda_i$ and $p_{i+1,i} = \mu_{i+1} > 0$ for every $i \in \{0, \dots, s-1\}$ has a unique stationary distribution π with*

$$\pi_i = \pi_0 \cdot \prod_{j=0}^{i-1} \frac{\lambda_j}{\mu_{j+1}} \qquad (2)$$

for all $i \in \{0, \dots, s\}$ where π_0 is chosen so that $\sum_{i=0}^{s} \pi_i = 1$.

PROOF. The following conditions must be satisfied by any stationary distribution π:

$$\begin{aligned} \lambda_0 \cdot \pi_0 &= \mu_1 \cdot \pi_1 \\ \lambda_{i-1}\pi_{i-1} + \mu_{i+1}\pi_{i+1} &= (\lambda_i + \mu_i)\pi_i \quad \forall i \in \{1, \dots, s-1\} \\ \lambda_{s-1} \cdot \pi_{s-1} &= \mu_s \cdot \pi_s \end{aligned}$$

Using these, equation (2) can easily be shown by induction on i, starting with $i = 1$. □

Hence, P has a unique stationary distribution π. In order to investigate how P approaches π when starting with the state 0 (i.e., all pebbles are initially white), we need the concept of domination.

DEFINITION 2.4. *Given two probability distributions q and q' on the state space U' we say that q* dominates *q', or $q \succeq q'$, if for all $i \in \{0, \dots, s\}$, $\sum_{j \geq i} q_j \geq \sum_{j \geq i} q'_j$.*

LEMMA 2.5. *Given any stochastic process $P = (p_{i,j})$ on some state space $U = \{0, \dots, s\}$, let $p_i = (p_{i,j})_{j \in U}$ denote the vector of transition probabilities for state i. If $p_i \succeq p_{i'}$ for every pair $(i, i') \in U$ with $i > i'$ then it holds for any two probability distributions q and q' on U: if $q \succeq q'$ then $q \cdot P \succeq q' \cdot P$.*

PROOF. Follows directly from the insight that q' can be produced from q by moving probability pieces to lower states. □

For any $i \geq 0$ let q_i denote the probability distribution of S after processing the ith join request. It certainly holds that $q_1 \succeq q_0$. Hence, it follows from Lemma 2.5 that $q_{t+1} \succeq q_t$ for every $t \geq 0$, implying that P monotonically converges against its stationary distribution π. In order to study this convergence in a rigorous way, we simplify P to the stochastic process P' with

$$p'_{i,j} = \begin{cases} i \cdot \frac{n}{N^2} & : \quad j = i-1 \\ (s-i) \cdot \frac{2N-(n+s)}{N^2} & : \quad j = i+1 \\ 1 - (p'_{i,i-1} + p'_{i,i+1}) & : \quad j = i \end{cases}$$

Consider now the following lemma.

LEMMA 2.6. *For any two stochastic processes P and P' on some state space $U = \{0, \dots, s\}$ with $p'_{i,j} \geq p_{i,j}$ for all $j < i$ and $p'_{i,j} \leq p_{i,j}$ for all $j > i$ it holds for any probability distribution q on U that $q \cdot P \succeq q \cdot P'$.*

PROOF. Consider any probability distribution q, and let $r = q \cdot P$ and $r' = q \cdot P'$. Then it holds for every $i \geq 1$ that

$$\begin{aligned} \sum_{j \geq i} r_j &= \sum_{j \geq i} q_j (1 - \sum_{k<i} p_{j,k}) + \sum_{j<i} q_j \sum_{k \geq i} p_{j,k} \\ &\geq \sum_{j \geq i} q_j (1 - \sum_{k<i} p'_{j,k}) + \sum_{j<i} q_j \sum_{k \geq i} p'_{j,k} \;=\; \sum_{j \geq i} r'_j \end{aligned}$$

and therefore $r \succeq r'$. □

Lemmas 2.5 and 2.6 imply that $q_0 \cdot P^t \succeq q_0 \cdot (P')^t$ for all $t \geq 0$. Hence, in order to obtain a lower bound for the worst-case number of black pebbles in S, it suffices to focus on P'. First, we investigate the stationary distribution of P'.

LEMMA 2.7. *Consider any birth-death process P on a state space $U = \{0, \dots, s\}$ with transition probabilities $p_{i,i+1} = \alpha(s-i)$ and $p_{i+1,i} = \beta(i+1)$ for every $i \in \{0, \dots, s-1\}$. Let the random variable X be the state of P. In the stationary distribution, $\mathrm{E}[X] = \alpha s/(\alpha+\beta)$. Moreover, for any $\epsilon \geq 0$,*

$$\Pr[X \geq (1+\epsilon)\mathrm{E}[X]] \leq e^{-\min[\epsilon,\epsilon^2]\mathrm{E}[X]/3}$$

and for any $\epsilon \in [0, 1]$,

$$\Pr[X \leq (1-\epsilon)\mathrm{E}[X]] \leq e^{-\epsilon^2 \mathrm{E}[X]/2}$$

PROOF. It is not difficult to show that the unique stationary distribution π satisfies

$$\pi_i = \binom{s}{i}\left(\frac{\alpha}{\alpha+\beta}\right)^i \left(\frac{\beta}{\alpha+\beta}\right)^{s-i}$$

for every $i \in \{0, \dots, s\}$. Hence, π has the same distribution as s independent Bernoulli trials with probability $p = \alpha/(\alpha+\beta)$, which implies that $\mathrm{E}[X] = \alpha s/(\alpha+\beta)$. The probability bounds for X immediately follow from the well-known Chernoff bounds. □

Let $N = (1+\epsilon)n$. It follows from Lemma 2.7 that P' has a unique stationary distribution π in which, on expectation, S has

$$\frac{2N-(n+s)}{2N-(n+s)+n} \cdot s = \frac{2N-n-s}{2N-s} \cdot s \approx \frac{1+2\epsilon}{2+2\epsilon} \cdot s$$

many black pebbles. It remains to bound the speed of convergence towards the stationary distribution. Given any probability distribution q on U', consider the potential function

$$\Phi = \sum_{i=1}^{s} |\pi_{\geq i} - q_{\geq i}|$$

where $\pi_{\geq i} = \sum_{j \geq i} \pi_j$ and $q_{\geq i} = \sum_{j \geq i} q_j$. Let $q' = q \cdot P'$. Suppose that $\pi \succeq q$. Then also $\pi \succeq q'$ by Lemma 2.5, and therefore

$$\begin{aligned} \Phi' &= \sum_{i=1}^{s} |\pi_{\geq i} - q'_{\geq i}| \;=\; \sum_{i=1}^{s} (\pi_{\geq i} - q'_{\geq i}) \\ &= \sum_{i=1}^{s} (\pi_{\geq i} - (q_{\geq i} + \lambda_{i-1}q_{i-1} - \mu_i q_i)) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{s}(\pi_{\geq i} - q_{\geq i}) - \sum_{i=1}^{s}\lambda_{i-1}q_{i-1} + \sum_{i=1}^{s}\mu_i q_i \\
&= \Phi - \sum_{i=0}^{s}(s-i)\frac{2N-(n+s)}{N^2}q_i + \sum_{i=0}^{s} i\frac{n}{N^2}q_i \\
&= \Phi - \frac{s}{N}\cdot\frac{2N-(n+s)}{N} + \frac{2N-s}{N^2}\sum_{i=1}^{s}q_{\geq i} \\
&= \Phi - \frac{2N-s}{N^2}\left(\frac{2N-n-s}{2N-s}s - \sum_{i=1}^{s}q_{\geq i}\right) .
\end{aligned}
$$

Now, notice that $\sum_{i=1}^{s}\pi_{\geq i} = \frac{2N-n-s}{2N-s}s$ and that $\sum_{i=1}^{s}q_{\geq i} = \sum_{i=1}^{s}\pi_{\geq i} - \Phi$. Hence, $\Phi' = \Phi - \frac{2N-s}{N^2}\Phi = (1 - \frac{2-s/N}{N})\Phi$. Because initially, $\Phi = \sum_{i=1}^{s}\pi_{\geq i} = \frac{2N-n-s}{2N-s}s$, it follows that it takes at most $O(N\log(s/\delta)) = O(n\log(s/\delta))$ steps to get δ-close to the stationary distribution. Thus, we get:

LEMMA 2.8. *If the adversary has at least s black pebbles, then it takes at most $O(n\log s)$ join requests on expectation and at most $O((n\log s)\log n)$ requests with high probability until the adversary has at least $s/2$ black pebbles in S.*

PROOF. Recall that $q_0P^t \succeq q_0(P')^t$ for any q_0 and $t \geq 0$. From our calculations above it follows that after $\Theta(n\log s)$ steps, $\sum_i|\pi_{\geq i} - q_{\geq i}| \leq 1/4$ and therefore also $|\pi_{\geq s/2} - q_{\geq s/2}| \leq 1/4$. Since π is a binomial distribution with expectation above $1/2$, it follows that $\pi_{\geq s/2} \geq 1/2$ and therefore $\Pr[S$ has $\geq s/2$ black pebbles$] \geq 1/4$. This probability bound holds irrespective of the state that S had $\Theta(n\log s)$ steps before, which completes the proof. □

Hence, the adversary can still win for $k = 2$. For $k = 3$ this will be much harder to achieve.

2.3 Outcome for $k = 3$

We start with a lower bound and then prove an almost matching upper bound on the worst-case number of black pebbles in a sequence of size s.

2.3.1 Lower bound

We start with a lower bound on the number of black pebbles in some sequence S of length s, using again the assumption that the adversary never evicts a black pebble from S. Given that S is in state i, it follows from the transition probabilities in Section 2.2 that after the join operation has finished, S has a probability distribution of $(q_{(i-1,0)}, q_{(i,0)}, q_{(i+1,0)}, q_{(i+2,0)})$ with

$$
\begin{aligned}
q_{(i-1,0)} &= \frac{i}{N}\cdot\frac{n-s+i}{N}\left(2-\frac{s-(i-1)}{N}\right) \\
q_{(i,0)} &= 1-(q_{(i-1,0)}+q_{(i+1,0)}+q_{(i+2,0)}) \\
q_{(i+1,0)} &= \frac{s-i}{N}\left(3-\frac{2n+i+1+n/N}{N}+\right. \\
&\qquad \left.\frac{(s-(i+1))(i+1)+2i(s-i)}{N^2}\right) \\
q_{(i+2,0)} &= \frac{s-(i+1)}{N}\cdot\frac{s-i}{N}\cdot\frac{N-n-(i+1)}{N}
\end{aligned}
$$

Hence, we can model the effect of a join operation on sequence S as a stochastic process P on the state space $U' = \{0,\ldots,s\}$ with transition probabilities

$$
p_{i,j} = \begin{cases}
\frac{i}{N}\cdot\frac{n-s+i}{N}\left(2-\frac{s-(i-1)}{N}\right) & : \quad j = i-1 \\
\frac{s-i}{N}\left(\frac{3N-(2n+i+1+n/N)}{N} + \ldots\right) & : \quad j = i+1 \\
\frac{s-(i+1)}{N}\cdot\frac{s-i}{N}\cdot\frac{N-n-(i+1)}{N} & : \quad j = i+2 \\
1-(p_{i,i-1}+p_{i,i+1}+p_{i,i+2}) & : \quad j = i
\end{cases}
$$

In order to obtain a lower bound on the number of black pebbles that can be in S in the worst case, we simplify the system P to a system P' with

$$
p'_{i,j} = \begin{cases}
i\cdot\frac{2n}{N^2} & : \quad j = i-1 \\
(s-i)\cdot\frac{3N-(2n+s+1)}{N^2} & : \quad j = i+1 \\
1-(p'_{i,i-1}+p'_{i,i+1}) & : \quad j = i
\end{cases}
$$

Since $p'_{i,i-1} \geq p_{i,i-1}$, $p'_{i,i+1} \leq p_{i,i+1}$, and $p'_{i,i+2} \leq p_{i,i+2}$, Lemmas 2.5 and 2.6 imply that the stationary distribution of P dominates the stationary distribution of P', as needed. Interpreting the probabilities as $p'_{i,i+1} = \alpha(s-i)$ and $p'_{i,i-1} = \beta\cdot i$, it follows from Lemma 2.7 that the expected value of the stationary distribution of P' is at least

$$
\frac{(3N-2n)/N^2\cdot s}{(3N-2n)/N^2+2n/N^2} = \frac{3N-2n}{3N}\cdot s = \frac{1+3\epsilon}{3+3\epsilon}\cdot s
$$

up to negligible terms. According to Lemma 2.7, it also holds that the number of black pebbles in S is eventually at least $(1-\delta)\frac{1+3\epsilon}{3+3\epsilon}\cdot s$, with high probability. Next we answer the question whether it can be much worse than that.

2.3.2 Upper bound

The upper bound is significantly more difficult to show because now we have to argue about *arbitrary* adversaries. At first glance, it may look like the stochastic process considered for the lower bound should be the worst-case process for arbitrary adversaries because it intuitively makes sense that the adversary should not remove a black pebble from a sequence S in which it tries to maximize the number of black pebbles. While this intuition will turn out to be correct, the process in the lower bound is not the worst possible because there is a subtle issue in the probabilities for stage k that allows us to construct a worse stochastic process. More precisely, if we use the rule that whenever a new position p' is created directly to the left of S and a white pebble is placed into it, p' is *not* included into S, we arrive at something worse. (The process studied in Section 2.2 *always* includes p' into S, no matter what pebble is placed into it. This follows from the definition of the sequences S_p.) Unfortunately, this new rule can create dependencies that are very hard to resolve (basically, Lemma 2.2 does not hold any more), but fortunately there is a way of working around this problem using combinatorial techniques.

Our proof for the upper bound proceeds as follows. First, we introduce a combinatorial technique allowing us to cover any sequence of s consecutive pebbles on the ring at any time and to model the outcome of adversarial behavior on them as a near-Markov chain under the assumption that the adversary never removes a black pebble from the considered sequence. By "near-Markov" we mean that transition probabilities only depend on the current state but may differ for different join requests. Afterwards, we show that this stochastic process dominates any stochastic process on any sequence under any adversarial behavior, completing the proof.

Counting all sequences

Consider the graph $G = (V, E)$ in which we have a node for every initial position on the ring and every new position created by a join request. There is a directed edge from v to w if and only if w is the successor of v on the ring when w is created. Every directed edge represents an *option* to follow a certain sequence in a sense that if we are currently considering S_v (i.e. the s closest successors of v on the ring) and w becomes the closest successor of v, we may either choose to continue with S_v (which means to include w into the old S_v) or to continue with S_w (which means just to stay with the old S_v).

Certainly, for any adversarial strategy, G is a forest, and every node w associated with a new position has a directed path from an initial node v. This path specifies a unique trajectory for a sequence originally starting with S_v. Let W be the set of nodes visited along the path from v to w in G (including v and w) and S_W be the sequence associated with it. We also say that W is a *witness* for S_W. The following crucial fact is easy to see.

FACT 2.9. *For every sequence S on the ring at any time there is a witness W so that $S = S_W$.*

Given a sequence S, let $b(S)$ denote the number of black pebbles in S. Because for any outcome with a sequence S so that $b(S) = b$ for some b it holds that there exists a W with $b(S_W) = b$, it holds for any b after any number of join requests executed by the adversary that

$$\Pr[\exists S:\ b(S) \geq b] \leq \sum_W \Pr[W] \cdot \Pr[b(S_W) \geq b \mid W]$$

where the sum sums up over all possible witnesses W and $\Pr[W]$ is the probability that W represents a directed path in G, i.e., S_W is well-defined.

Some of the witnesses are very unlikely to be true, and we therefore want to remove them from consideration for the rest of the proof. Suppose that the initial positions have numbers from 1 to n and the position created by the jth join request has number $n + j$. Thus, any witness W is a subset of $M = \{1, \ldots, n+m\}$. Let I be the set of all sets $\{i, \ldots, i + c \log n\} \subset M$ for some fixed constant $c > 3$. Given any numbers $p, p' \in M$ with $p < p'$, the probability that p' is the successor of p when created is at most $1/n$. Hence, we get:

$$\begin{aligned}
&\Pr[\exists W \subseteq M:\ W \text{ is true and } \exists J \in I:\ |W \cap J| \geq c] \\
&\quad\leq\quad (n+m)\binom{c \log n}{c}\left(\frac{1}{n}\right)^{c-1} \\
&\quad\leq\quad (n+m)n \cdot \left(\frac{e \log n}{n}\right)^{c-1} \leq n^{-c+3}
\end{aligned}$$

Thus, we can ignore in the following witnesses that are too dense (i.e., that satisfy the property $|W \cap J| \geq c$). In the following, let $\mathcal{W}$ be the set of all sparse witnesses.

Because for any outcome of m join requests there are exactly $n+m$ witnesses that are well-defined (note that a witness may just contain an initial position), it holds that $\sum_W \Pr[W] = n + m$. Thus, if we can show for some b that

$$\max_{W \in \mathcal{W}} \Pr[b(S_W) \geq b \mid W] \leq \frac{p}{n+m}$$

for some $p \in [0, 1]$, then $\Pr[\exists S:\ b(S) \geq b] \leq p + n^{-c+3}$. Our goal will be to find a b so that p is polynomially small in n so that for a polynomial number of join requests it is very improbable that there ever exists a sequence S with $b(S) \geq b$.

Adversaries without S-departures

Consider now some fixed witness $W \in \mathcal{W}$. Let P_k be the transition matrix of the k-rotation rule as implied by the transition probabilities for the various stages in Section 2.2. P_2 and P_3 have already been derived above. We use W to define a stochastic process on a sequence S that is initially equal to S_p where p is the smallest element (i.e., an initial position) in W. S is initially in state 0. Afterwards, we process the numbers $n + 1, \ldots, m$ in a consecutive way starting with $n + 1$. For each number $j \in \{n + 1, \ldots, m\}$, we check whether $j \in W$. If $j \notin W$, then we apply P_3 to S, and if $j \in W$, we apply P_2 to S. This has the following justification:

If $j \notin W$, then we do not assume anything about the new position of j because we will continue to follow the sequence associated with the current position, and hence we can use the transition matrix P_3 because we are using the 3-rotation rule. If $j \in W$, however, then the new position for j is predetermined to be the new successor of the current position considered by us. Hence, the transition probabilities in stage 3 are not applicable to the current sequence S. Using only the transition probabilities of stages 0 to 2 results in the transition matrix P_2. If we assume now that the adversary never removes a pebble from S, then none of the two cases creates a bias in the pebble distribution in S, and therefore Lemma 2.2 holds at any time, which implies that the use of P_3 for $j \notin W$ is indeed correct.

It remains to bound the probability distribution of S after all requests have been processed. For this, we first assume that *only* P_3 is applied to S. In order to obtain an upper bound on the number of black pebbles in S, we simplify the transition probabilities in P_3 to

$$p'_{i,j} = \begin{cases} i \cdot \frac{n-s}{N^2}\left(2 - \frac{s}{N}\right) & :\ j = i-1 \\ (s-i) \cdot \frac{3N-2n}{N^2} & :\ j = i+1 \\ (s-i-1)(s-i) \cdot \frac{N-n}{N^3} & :\ j = i+2 \\ 1 - (p'_{i,i-1} + p'_{i,i+1} + p'_{i,i+2}) & :\ j = i \end{cases} \tag{3}$$

Since $p'_{i,i-1} \leq p_{i,i-1}$, $p'_{i,i+1} \geq p_{i,i+1}$, and $p'_{i,i+2} \geq p_{i,i+2}$ for any $N \geq n$, Lemmas 2.6 and 2.5 imply that the stationary distribution of P' dominates the stationary distribution of P_3, as needed.

Also, notice that the *current* N used in the system may differ from join operation to join operation, depending on how many black pebbles the adversary currently has in the system. However, it is easy to check that for any two values N_1, N_2 with $N_1 < N_2$, $p'_{i,i-1}(N_1) \geq p'_{i,i-1}(N_2)$, $p'_{i,i+1}(N_1) \leq p'_{i,i+1}(N_2)$, and $p'_{i,i+2}(N_1) \leq p'_{i,i+2}(N_2)$. Hence, as a worst case, we can assume that N is maximal possible in each join operation.

Now, recall that when starting in the state 0 (only white pebbles are in the sequence S), the stationary distribution π of P' dominates the probability distribution of S at any time. Hence, all we need to do for an upper bound on the worst-case distribution of S is to find a probability distribution q that dominates π.

In the following, let $\alpha = (3N - 2n)/N^2$, $\beta = (2 - s/N)(n - s)/N^2$, $\gamma = (N - n)/N^3$, and $\alpha' = (1 + \delta)\alpha$ for some $\delta > 0$ determined later. Consider the distribution q with

$$q_i = \binom{s}{i}\left(\frac{\alpha'}{\alpha' + \beta}\right)^i \left(\frac{\alpha'}{\alpha' + \beta}\right)^{s-i}$$

for all $i \in \{0, \ldots, s\}$. Then it holds for $q' = q \cdot P'$ that

$$\begin{aligned}
q'_{\geq i} &= q_{\geq i} - \beta i \cdot q_i + \alpha(s - i - 1)q_{i-1} + \\
&\quad\ \gamma(s - i + 2)(s - i + 1)q_{i-2} \\
&= q_{\geq i} - \beta i \cdot q_i + \frac{\alpha \cdot \beta}{\alpha'} i \cdot q_i + \gamma\left(\frac{\beta}{\alpha'}\right)^2 i(i-1)q_i \\
&= q_{\geq i} - \left(\frac{\delta}{1+\delta} - (i-1)\frac{\gamma\beta}{((1+\delta)\alpha)^2}\right)\beta i \cdot q_i
\end{aligned}$$

for all $i \geq 1$. Notice that $\gamma\beta/\alpha^2 \leq 1/N$. Hence, $q'_{\geq i} \leq q_{\geq i}$ for all $i \geq 1$ if δ is chosen so that $\delta/(1+\delta) - s/((1+\delta)^2 N) \geq 0$, which is true for $\delta \geq s/N$. Hence, for this δ, $q \succeq q \cdot P$, and because of Lemma 2.5 this implies that $q \succeq q \cdot P^t$ for any $t \geq 0$. Since $q \cdot P^t$ converges against the stationary distribution π of P', it follows that $q \succeq \pi$. Hence, according to Lemma 2.7, it follows that the expected number of black pebbles in S in the stationary distribution, when using only P_3, is at most

$$\frac{\alpha'}{\alpha' + \beta} s = \frac{1+3\epsilon}{3+3\epsilon} s$$

up to negligible terms. It remains to show this result cannot be perturbed too much by P_2-transitions. For this we need three lemmas. Recall that $\binom{n}{k} = [n]_k/k!$ with $[n]_k = n!/(n-k)!$. For any $k > n$, $[n]_k = 0$ and therefore also $\binom{n}{k} = 0$. By definition, $\binom{0}{0} = 1$.

LEMMA 2.10. *Consider any two birth-death processes P and P' on a state space $U = \{0, \ldots, s\}$ with transition probabilities $p_{i,i+1} = \alpha(s-i)$, $p_{i+1,i} = \beta(i+1)$, $p'_{i,i+1} = \alpha'(s-i)$, and $p'_{i+1,i} = \beta'(i+1)$ for all $i \in \{0, \ldots, s-1\}$. Any product of matrices out of P and P' in which P appears t times and P appears t' times results in a stochastic matrix P'' with*

$$p''_{i,i-d} \geq [i]_d (1 - s \cdot m)^d \cdot f_d(\min[\beta, \beta'], \max[\beta, \beta'])$$

and

$$p''_{i,i+d} \leq [s-i]_d \cdot f_d(\max[\alpha, \alpha'], \min[\alpha, \alpha'])$$

for all $d \geq 1$ where $m = \max[\alpha, \alpha', \beta, \beta']$. $f_1(x,y) = t_x x + t_y y$ with $t_x = t$ if $x \in \{\alpha, \beta\}$ and otherwise $t_x = t'$, and for all $d \geq 2$,

$$\begin{aligned} f_d(x,y) &= \left(\binom{t_x + t_y}{d} - \binom{t_y}{d} \right) \cdot x^{\min[d, t_x]} y^{\max[d - t_x, 0]} \\ &\quad + \binom{t_y}{d} \cdot y^d \end{aligned}$$

PROOF. Similar to the proof of Lemma 2.15. □

LEMMA 2.11. *For any set of constants $k \geq 1, \gamma_1, \ldots, \gamma_k \geq 0$ and $\alpha'', \beta'' \in (0,1)$, the transition matrix $Q = (q_{i,j})$ on $U = \{0, \ldots, s\}$ with $q_{i,j} = \gamma_{j-i} (\alpha'')^{j-i} [s-i]_{j-i}$ for any $j > i$ and $q_{i,j} = \gamma_{i-j} (\beta'')^{i-j} [i]_{i-j}$ for any $j < i$ has a unique stationary distribution π with*

$$\pi_i = \binom{s}{i} \left(\frac{\alpha''}{\alpha'' + \beta''} \right)^i \left(\frac{\beta''}{\alpha'' + \beta''} \right)^{s-i}$$

PROOF. Can easily be checked by computing $\pi \cdot P$. □

LEMMA 2.12. *Let the values in Lemma 2.10 be defined as $\alpha = (1+s/N)(3N-2n)/N^2$, $\beta = (2-s/N)(n-s)/N^2$, $\alpha' = (2N-n)/N^2$, $\beta' = (n-s)/N^2$, $t = c(\log n - 1)$, and $t' = c$. Let the values in Lemma 2.11 be given as $\alpha'' = \alpha$, $\beta'' = (1 - 2/\log n)\beta$, and $\gamma_d = \binom{t+t'}{d}$, $1 \leq d \leq t+t'$. Then it holds that $q_{i,j} \leq p''_{i,j}$ for all $j < i$ and $q_{i,j} \geq p''_{i,j}$ for all $j > i$.*

PROOF. It is easy to verify that $[s-i]_d \gamma_d (\alpha'')^d \geq p''_{i,i+d}$ for all $d \geq 1$. To see that $[i]_d \gamma_d (\beta'')^d \leq p''_{i,i-d}$, one has to distinguish between $d \leq (c/2)\log n$ and $d \geq (c/2)\log n$ and use the fact that $f(\beta', \beta) \geq \max[\binom{t+t'}{d} (\beta')^c \beta^{d-c}, \binom{t}{d}\beta]$. □

Thus, the worst-case stationary distribution of our stochastic process with P_2- and P_3-transitions is dominated by a binomial distribution with an expected value of at most $(1+3\epsilon)/(3+3\epsilon)s + O(1)$. Also, due to Lemma 2.7, the probability that there are more than $(1+\delta)\frac{1+3\epsilon}{3+3\epsilon} \cdot s$ black pebbles in S is polynomially small in n for any constant $\delta > 0$ if $s = O(\log n)$ is sufficiently large, completing the proof for adversaries without S-departures.

Arbitrary adversaries

The departure of a black pebble can certainly never increase the number of black pebbles in S, even if the new pebble added to the right is black, but it can create a bias towards which pebbles are evicted in stage k. To handle this bias, we compare the pebble distribution in $S = S_W$ for any W due to adversarial strategies *without* S-departures with adversarial strategies *with* S-departures.

Consider any (adaptive or non-adaptive) adversarial strategy A, and let A^* be A in which any leave request of a black pebble in S is replaced by a leave request of any black pebble outside of S. To make sure that this is always possible, we assume for A^* that initially s randomly selected white pebbles are converted into black pebbles. In this way, A^* will never run out of black pebbles outside of S, no matter what A is doing. Our aim will be to show that at any time, the probability distribution over the number of black pebbles in S w.r.t. A^* dominates the probability distribution over the number of black pebbles in S w.r.t. A. This allows us to reduce arbitrary adversaries to adversaries that never remove a black pebble from S, as desired.

A *configuration* of S is represented by a tuple (v, m) where $v \in \{0,1\}^s$ represents the distribution of white and black pebbles in S (0:white, 1:black) and m denotes the total number of black pebbles in the system. Let Γ be the set of all possible configurations of S, and let $p_j : \Gamma \rightarrow [0,1)$ (resp. $p^*_j : \Gamma \rightarrow [0,1)$) be the probability distribution over Γ after the jth request of A (resp. A^*). Given two configurations $C = (v, m)$ and $C' = (v', m')$, we say that C *dominates* C', or $C \succeq C'$, if and only if $v \succeq v'$ (which is defined here as $\sum_{j=1}^{i} v_j \geq \sum_{j=1}^{i} v'_j$ for all i) and $m = m' + s$. We will show the following lemma.

LEMMA 2.13. *For any adversarial strategy A, there is a probability distribution $q_j : \Gamma^2 \rightarrow [0,1)$ for every $j \geq 0$ so that $C^* \succeq C$ for every pair $(C, C^*) \in \Gamma^2$ with $q_j(C, C^*) > 0$, $\sum_{C^* \in \Gamma} q_j(C, C^*) = p_j(C)$ for all $C \in \Gamma$, and $\sum_{C \in \Gamma} q_j(C, C^*) = p^*_j(C^*)$ for all $C^* \in \Gamma$.*

PROOF. We prove the lemma by induction on j. For $j = 0$, the lemma is certainly true because for A there are initially only white pebbles in S. So let us assume that for some $j \geq 0$ the lemma is already true. Then we will show that it is also true for $j+1$. The $(j+1)$st operation of A can be either a join request or a leave request.

Join requests. First, suppose that it is a join request. We analyze the effect of such a request by introducing *transitional configurations*. A *transitional configuration* C is a tuple $(v, h, m) \in \{0,1\}^{s+1} \times \mathbb{N}_0$ where v represents the pebble distribution in S, h represents the state of the homeless pebble, and m is the total number of black pebbles in the system. Let Ω be the set of all transitional configurations. For two configurations $C, C' \in \Omega$ we say that $C \succeq C'$ if and only if $v \succeq v'$, $h \geq h'$, and $m = m' + s$.

CLAIM 2.14. *Given any two transitional configurations $C, C^* \in \Omega$ with $C^* \succeq C$ at the beginning of stage t of the join request, it holds at the end of stage t: there is a probability distribution $q : \Omega^2 \rightarrow [0,1)$ so that $D^* \succeq D$ for every pair $(D, D^*) \in \Omega^2$ with $q(D, D^*) > 0$, $\sum_{D^* \in \Omega} q_j(D, D^*) = \Pr_A[D]$ for all $D \in \Omega$, and $\sum_{D \in \Omega} q(D, D^*) = \Pr_{A^*}[D^*]$ for all $D^* \in \Omega$.*

PROOF. Consider any pair $(C, C^*) \in \Omega^2$ with $C^* \succeq C$. Let $C = (v, h, m)$ and $C^* = (v^*, h^*, m^*)$. If $t = k$, i.e., the homeless pebble is inserted into a new position, then it is easy to check that for any position i in which the homeless pebble is placed in v and

v^*, the outcome w of v and the outcome of w^* of v^* satisfy $w^* \succeq w$. Hence, it remains to consider the case $t < k$.

Consider any fixed extension of v to the pebbles outside of S and any fixed extension of v^* in C^* to the pebbles outside of S so that v and v^* are both elements of $\{0,1\}^{n+m}$. We require a permutation π on $\{1,\dots,n+m\}$ satisfying the following conditions:

For all i, $\pi(\pi(i)) = i$ (i.e., π is a set of transpositions); for all i, $v_i^* \geq v_{\pi(i)}$; for all $i \leq s$ with $v_i^* < v_i$, $\pi(i) < i$; and for all $i \leq s$ with $\pi(i) < i$, $v^*_{\pi(i)} > v_{\pi(i)}$.

Since the positions 1 to s belong to S, $v^* \succeq v$, and $m^* = m+s$, all conditions can be satisfied.

For any $i \in \{1,\dots,n+m\}$ let $w^*(i)$ be the outcome of replacing the pebble in position i of v^* and w(i) be the outcome of replacing the pebble in position $\pi(i)$ of v. Then it follows from our definition of π that $w^*(i) \succeq w(i)$ (w.r.t. the first s positions) for every i. Also for the two new homeless pebbles $h^*(i)$ and $h(i)$ it holds that $h^*(i) \geq h(i)$. Since every position i is equally likely to be selected for the replacement, the claim holds also for $t < k$. □

It immediately follows from the claim that join requests satisfy the lemma.

Leave requests. Suppose that the $(j+1)$st operation of A is a leave request. Consider any two configurations $C = (v,m)$ and $C^* = (v^*,m^*)$ with $C \succeq C^*$, and let D resp. D^* be the configurations after executing the leave request of A resp. A^*. If A removes a black pebble outside of S, then it follows that $D = (v, m-1)$ and $D^* = (v^*, m^*-1)$, and hence $D^* \succeq D$. Otherwise, a black pebble is removed from v but no black pebble is removed from v^*. In this case, it is easy to verify that $v^* \succeq v$ and therefore also $D^* \succeq D$.

Hence, the induction step is true, which concludes the proof of the lemma. □

Given any adversarial strategy A, let the random variable X_j (resp. X_j^*) denote the number of black pebbles in S after the jth join request of A (resp. A^*). Lemma 2.13 immediately implies that $\Pr[X_j \geq b] \leq \Pr[X_j^* \geq b]$ for any j and any $b \in \{1,\dots,s\}$. Hence, it suffices for our upper bound to focus only on adversarial strategies that never remove a black pebble from S. Since removals of pebbles outside of S do not affect S, we can ignore these and therefore only need to study the effect of join requests on S. Hence, our analysis above for the case that an adversary never removes a pebble from S captures indeed the worst case that can happen to a sequence, which finishes the proof.

2.4 Outcome for $k > 3$

We only sketch the upper bound for all even $k > 3$, but using the techniques of the previous section, it is not difficult to extend it to all $k \geq 3$. Also a matching lower bound can be shown by providing a counterpart of Lemma 2.15 with slightly changed bounds.

Consider some fixed sequence S of length s. Since Lemma 2.13 holds for arbitrary k, it suffices for the worst possible distribution of black pebbles in S to study only adversaries that remove black pebbles outside of S. Hence, we only need to focus on the effect of join operations on S. The following lemma bounds the transition probabilities of the k-rotation rule. As before, let N denote the maximum number of pebbles in the system.

LEMMA 2.15. *For any $k > 3$, it holds for the transition matrix $P = (p_{i,j})$ of the k-rotation rule that for all $d \geq 1$,*

$$p_{i,i-d} \geq \frac{[i]_d}{N^d}\left(1 - \frac{s \cdot k}{N}\right)^d \left(\frac{n-ks}{N}\right)^d \binom{k-d}{d}$$

and

$$p_{i,i+d} \leq \frac{[s-i]_d}{N^d}\left(\frac{N-n}{N}\right)^{d-1}\left(\binom{k-d}{d-1} + \frac{N-n}{N}\binom{k-d}{d}\right)$$

PROOF. Given that the system starts in state $(i,0)$, it can be shown by induction that for every $t \geq 1$, the probability distribution $q^{(t)}$ at the beginning of stage t satisfies $q^{(t)}_{(i,0)} \geq \frac{n-ks}{N}$ and $q^{(t)}_{(i,1)} \leq \frac{N-n}{N}$. Hence, the probability moved from $(i,1)$ to $(i+1,0)$ is at most $\frac{s-i}{N}$ at stage 1 and at most $\frac{s-i}{N} \cdot \frac{N-n}{N}$ at all subsequent stages. Using this as the basis, it can be shown by induction on $d \geq 2$ that at stage $t \geq 2d-1$, a probability of at most

$$\frac{[s-i]_d}{N^d}\left(\frac{N-n}{N}\right)^{d-1}\left(\binom{t-d-1}{d-2} + \frac{N-n}{N}\binom{t-d-1}{d-1}\right)$$

is moved from $(i+d-1,1)$ to $(i+d,0)$. Summing up over all t gives the bound on $p_{i,i+d}$.

For the probability flow to lower states, note that for any flow from $(j,0)$ to $(j-1,1)$, at most a fraction of $\frac{sk}{N}$ will leave the states $(j-1,0)$ and $(j-1,1)$ over the remaining stages. Hence, by induction on $d \geq 1$, it can be shown that at any stage $t \geq 2d$, at least a probability of

$$\frac{[i]_d}{N^d}\left(1 - \frac{s \cdot k}{N}\right)^d \left(\frac{n-ks}{N}\right)^d \binom{t-d-1}{d-1}$$

will flow from $(i-d+1,0)$ to $(i-d,1)$ that will stay at $(i-d,0)$ or $(i-d,1)$. Summing up over these probabilities gives the bound on $p_{i,i-d}$. □

Now, consider the transition matrix $P' = (p'_{i,j})$ with

$$p'_{i,j} = \begin{cases} \frac{i}{N} \cdot \left(1 - \frac{sk}{N}\right) \frac{n-ks}{N}(k-1) & : \quad j = i-1 \\ \frac{s-i}{N} \cdot \left(1 + \frac{N-n}{N}(k-1)\right) & : \quad j = i+1 \\ 1 - (p'_{i,i-1} + p'_{i,i+1}) & : \quad j = i \end{cases}$$

Our goal will be to show that for any even $k > 3$, the stationary distribution of P' dominates the stationary distribution of P. In order to do this, we need the following lemma.

LEMMA 2.16. *Let $\alpha'' = \frac{1}{N}(1 + \frac{N-n}{N}(k-1))$ and $\beta'' = \frac{1}{N}(1 - \frac{ks}{N})\frac{n-ks}{N}(k-1)$ and let $\gamma_d = \binom{k-d}{d}/(k-1)^d$ for all $d \geq 1$. Then it holds for the system Q in Lemma 2.11 that $q_{i,j} \leq p_{i,j}$ for all $j < i$ and $q_{i,j} \geq p_{i,j}$ for all $j > i$.*

PROOF. It is easy to verify that $\gamma_d(\beta'')^d[i]_d \leq p_{i,i-d}$ for all $d \geq 1$. Furthermore, $\gamma_d(\alpha'')^d[s-i]_d \geq p_{i,i+d}$ for all $d \geq 1$ because

$$\begin{aligned} & \gamma_d\left(1 + \frac{N-n}{N}(k-1)\right)^d \\ \geq \quad & \gamma_d\left(\binom{d}{1}\left(\frac{N-n}{N}(k-1)\right)^{d-1} + \left(\frac{N-n}{N}(k-1)\right)^d\right) \\ = \quad & \left(\frac{N-n}{N}\right)^{d-1}\left(\binom{k-d}{d-1} + \frac{N-n}{N}\binom{k-d}{d}\right) \end{aligned}$$

□

Combining this with Lemmas 2.11 and 2.16 implies that the stationary distribution of P' indeed dominates the stationary distribution of P. Thus, it follows from Lemma 2.7 and the arguments in

Section 2.3 that the expected number of black pebbles in a sequence S of length s is at most

$$\frac{kN-(k-1)n}{kN-(k-1)n+(k-1)n} \cdot s + O(1) = \frac{1+k\epsilon}{k+k\epsilon} \cdot s + O(1)$$

Also, following along the lines of the proof for $k = 3$, the probability that there are more than $(1+\delta)\frac{1+k\epsilon}{k+k\epsilon}s$ black pebbles in S is polynomially small in n for any constant $\delta > 0$ if $s = O(\log n)$ is sufficiently large.

3. CONCLUSIONS

In this paper we only showed how to make a ring robust against adaptive join-leave attacks. Further research is needed to investigate strategies that can also handle join-leave attacks for other types of networks. For example, many peer-to-peer systems are based on the concept of virtual space, for which the k-rotation rule does not work because the node IDs have to be kept well-spread in the virtual space. As another example, if we want to maintain a dynamic random graph, we do not just have to make sure that each honest node has only a small constant fraction of its edges to adversarial nodes, with high probability, but we also have to make sure that those edges that it has to honest nodes are still sufficiently random.

Acknowledgements

I would like to thank Baruch Awerbuch for suggesting to me to work on scalable, proactive security mechanisms for peer-to-peer systems and for many helpful discussions.

4. REFERENCES

[1] I. Abraham and D. Malkhi. Probabilistic quorums for dynamic systems. In *Proc. of the 18th Annual Conference on Distributed Computing (DISC)*, 2003.

[2] J. Aspnes and G. Shah. Skip graphs. In *Proc. of the 14th ACM Symp. on Discrete Algorithms (SODA)*, pages 384–393, 2003.

[3] B. Awerbuch and C. Scheideler. Group Spreading: A protocol for provably secure distributed name service. In *Proc. of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, 2004.

[4] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorem for non-cryptographic fault tolerant distributed computing. In *Proc. of the 20th ACM Symp. on Theory of Computing (STOC)*, 1988.

[5] D. Boneh and M. Franklin. Efficient generation of shared RSA keys. In *Proc. of CRYPTO 97*, pages 425–539, 1997.

[6] C. Cachin, K. Kursawe, A. Lysyanskaya, and R. Strobl. Asynchronous verifiable secret sharing and proactive cryptosystems. In *Proc. of the 9th ACM Conference on Computer and Communications Security (CCS)*, 2002.

[7] R. Canetti, R. Gennaro, A. Herzberg, and D. Naor. Proactive security: Long-term protection against break-ins. *RSA CryptoBytes*, 3(1):1–8, 1997.

[8] R. Canetti, S. Halevi, and A. Herzberg. Maintaining authenticated communication in the presence of break-ins. *Journal of Cryptology*, 13(1):61–106, 2000.

[9] R. Canetti and A. Herzberg. Maintaining security in the presence of transient faults. In *Proc. of CRYPTO 94*, pages 425–438, 1994.

[10] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach. Security for structured peer-to-peer overlay networks. In *Proc. of the 5th Usenix Symp. on Operating Systems Design and Implementation (OSDI)*, 2002.

[11] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proc. of the 26th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 383–395, 1986.

[12] C.S. Chow and A. Herzberg. Network randomization protocol: A proactive pseudo-random generator. In *Proc. of the 5th USENIX UNIX Security Symposium*, pages 55–63, 1995.

[13] S. Crosby and D. Wallach. Denial of service via algorithmic complexity attacks. In *Usenix Security*, 2003.

[14] P. Diaconis and M. Shahshahani. Generating a random permutation with random transpositions. *Z. Wahrscheinlichkeitstheorie verw. Gebiete*, 57:159–179, 1981.

[15] J. R. Douceur. The sybil attack. In *Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[16] P. Druschel and A. Rowstron. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, 2001. See also `http://research.microsoft.com/∼antr/Pastry`.

[17] Y. Frankel, P. Gemmell, P. MacKenzie, and M. Yung. Proactive RSA. In *Proc. of CRYPTO 97*, 1997.

[18] Y. Frankel, P. Gemmell, P. D. MacKenzie, and M. Yung. Optimal resilience proactive public-key cryptosystems. In *Proc. of the 38th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 384–393, 1997.

[19] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. The round complexity of verifiable secret sharing and secure multicast. In *Proc. of the 33rd ACM Symp. on Theory of Computing (STOC)*, pages 580–589, 2001.

[20] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. *Information and Computation*, 164(1):54–84, 2001.

[21] R. Gennaro, M. Rabin, and T. Rabin. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In *Proc. of the 17th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 101–111, 1998.

[22] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing, or: How to cope with perpetual leakage. In *Proc. of CRYPTO 95*, pages 339–352, 1995.

[23] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung. Proactive public key and signature systems. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*, pages 100–110, 1997.

[24] U. Nadav and M. Naor. Fault tolerant storage in a dynamic environment. In *Proc. of the 18th Annual Conference on Distributed Computing (DISC)*, 2004.

[25] M. Naor and U. Wieder. Scalable and dynamic quorum systems. In *Proc. of the 22nd ACM Symp. on Principles of Distributed Computing (PODC)*, 2003.

[26] S. Nielson, S. Crosby, and D. Wallach. Kill the messenger: A taxonomy of rational attacks. In *Proc. of the 4th International Workshop on Peer-to-Peer Systems (IPTPS)*, 2005.

[27] R. Ostrovsky and M. Yung. How to withstand mobile virus attacks. In *Proc. of the 10th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 51–59, 1991.

[28] T. Rabin. A simplified approach to threshold and proactive RSA. In *Proc. of CRYPTO 98*, 1998.

[29] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. of the ACM SIGCOMM '01*, 2001.

[30] C. Scheideler. *Probabilistic Methods for Coordination Problems*. HNI-Verlagsschriftenreihe 78, University of Paderborn, 2000. See also `http://www.cs.jhu.edu/∼scheideler`.

[31] E. Sit and R. Morris. Security considerations for peer-to-peer distributed hash tables. In *Proc. of 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[32] M. Srivatsa and L. Liu. Vulnerabilities and security threats in structured overlay networks: A quantitative analysis. In *Proc. of the 20th IEEE Computer Security Applications Conference (ACSAC)*, 2004.

[33] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proc. of the ACM SIGCOMM '01*, 2001. See also `http://www.pdos.lcs.mit.edu/chord/`.

[34] B.Y. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report, UCB/CSD-01-1141, University of California at Berkeley, 2001. See also `http://www.cs.berkeley.edu/∼ravenben/tapestry`.