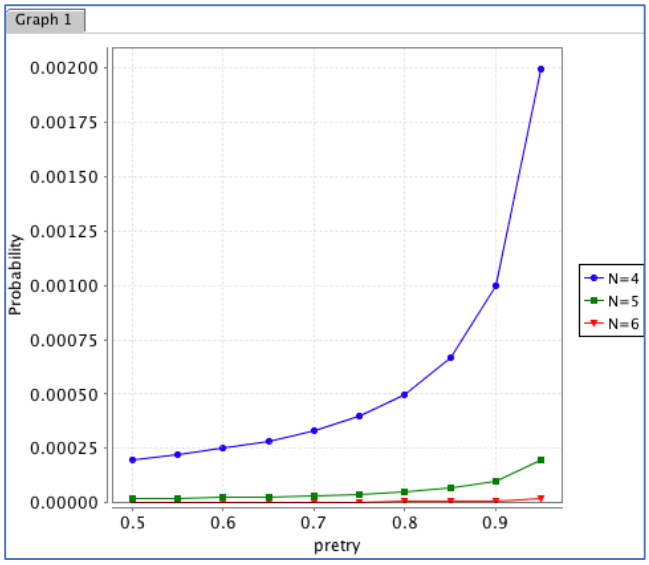
PSEC – Practical 2 Solutions

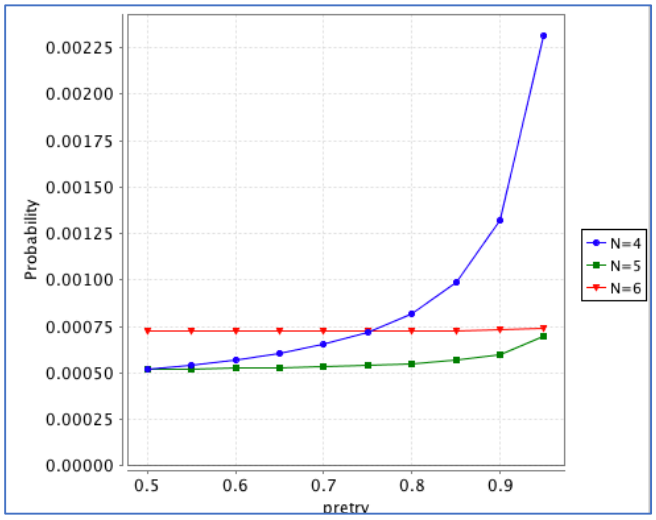
Exercise 1

Task 2

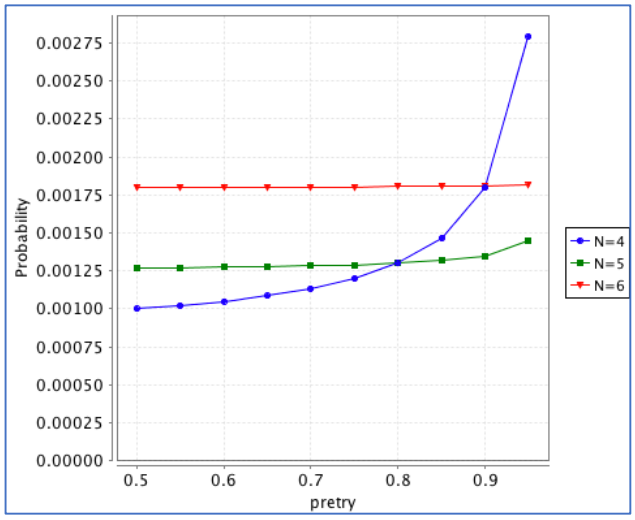


Task 3

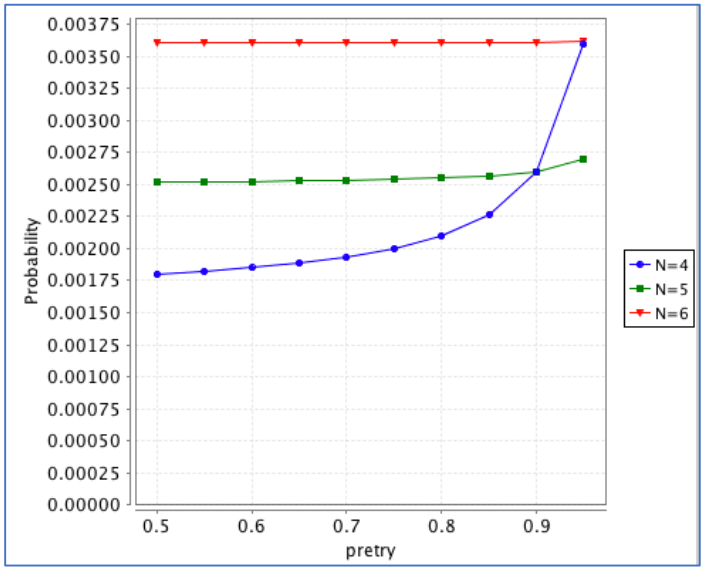
p=0.02



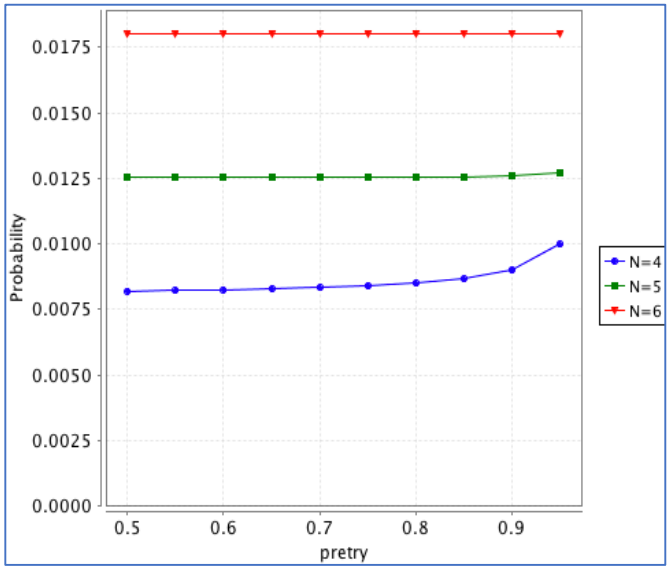
p=0.05



p=0.1



p=0.5



PSEC – Practical 2 Solutions

Task 4

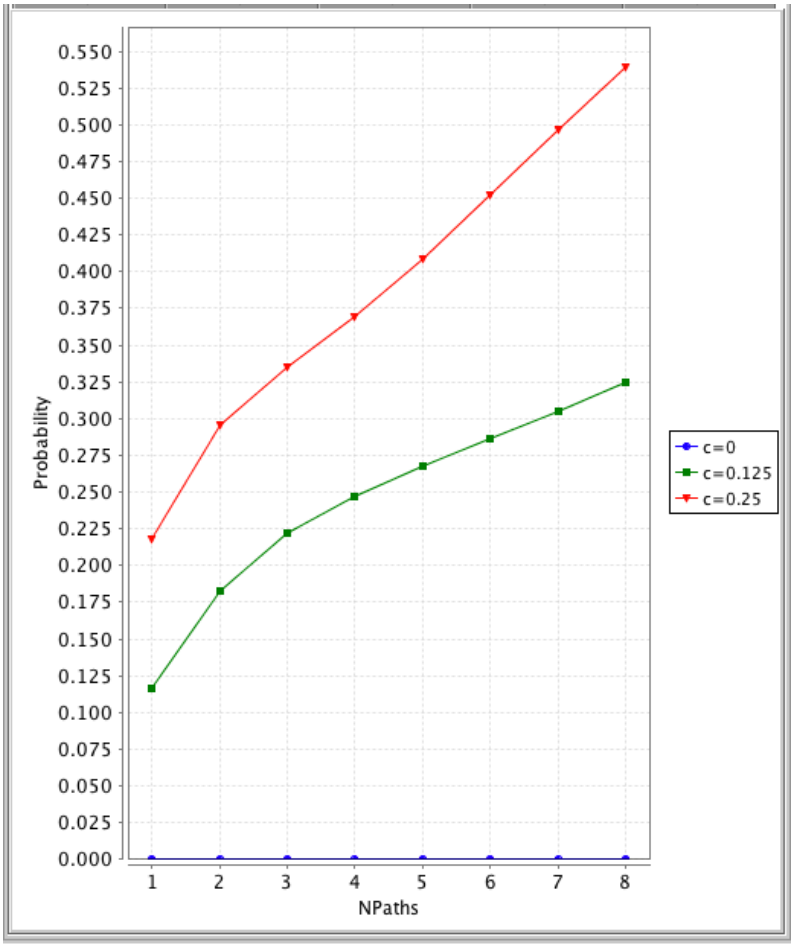
The probability that the PIN is obtained by the criminal increases as *pretry* becomes larger, so limit the number of retries permitted after failed attempts. Perhaps advise customers not to write the PIN down (in an obvious way) so *pwrite* and *pnotice* are reduced.

Exercise 2

Note: "However, the protocol has a vulnerability. When parties on an established path leave the system […] new paths need to be established." refers to an ongoing session between a web browser and a website, which normally uses a fixed path, having to find a new path.

Task 1

Analysis of P=? [F path=NPaths & log1>log2 & log1>log3 & log1>log4 & log1>log5 & log1>log6 & log1>log7 & log1>log8]

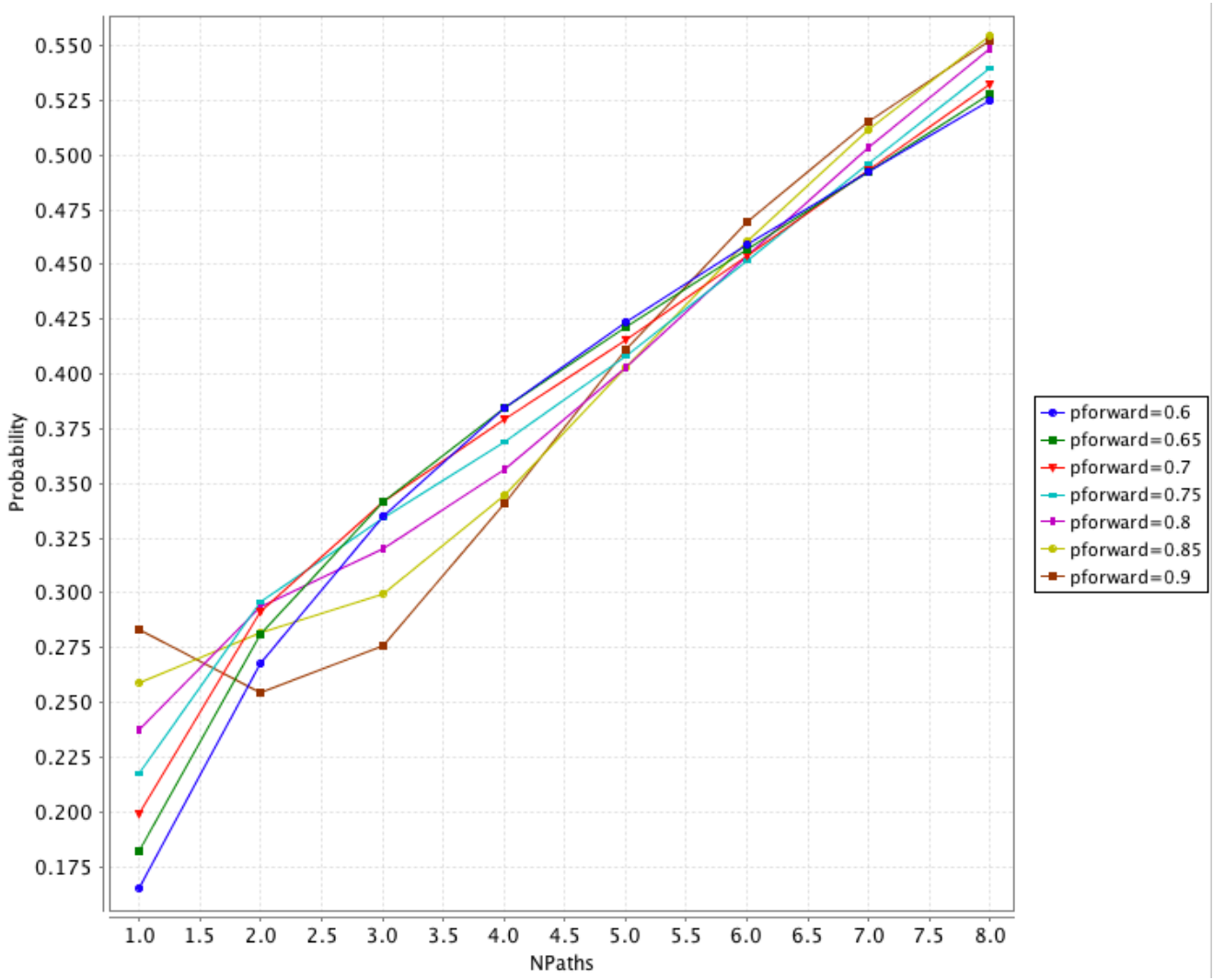


c=0 (no dishonest party) → 0 probability that the originator "stands out"

c=0.125 (1 dishonest party) → There is a pforward/8 probability that the originator sends the request to the dishonest party the first time, plus it may send the request to the dishonest part after one or several cycles of sending the request to itself! Probability of standing out increases non-linearly with the number of paths.

c=0.25 (2 dishonest parties) → similar behaviour as for c=0.125, but higher probabilities for all NPaths values

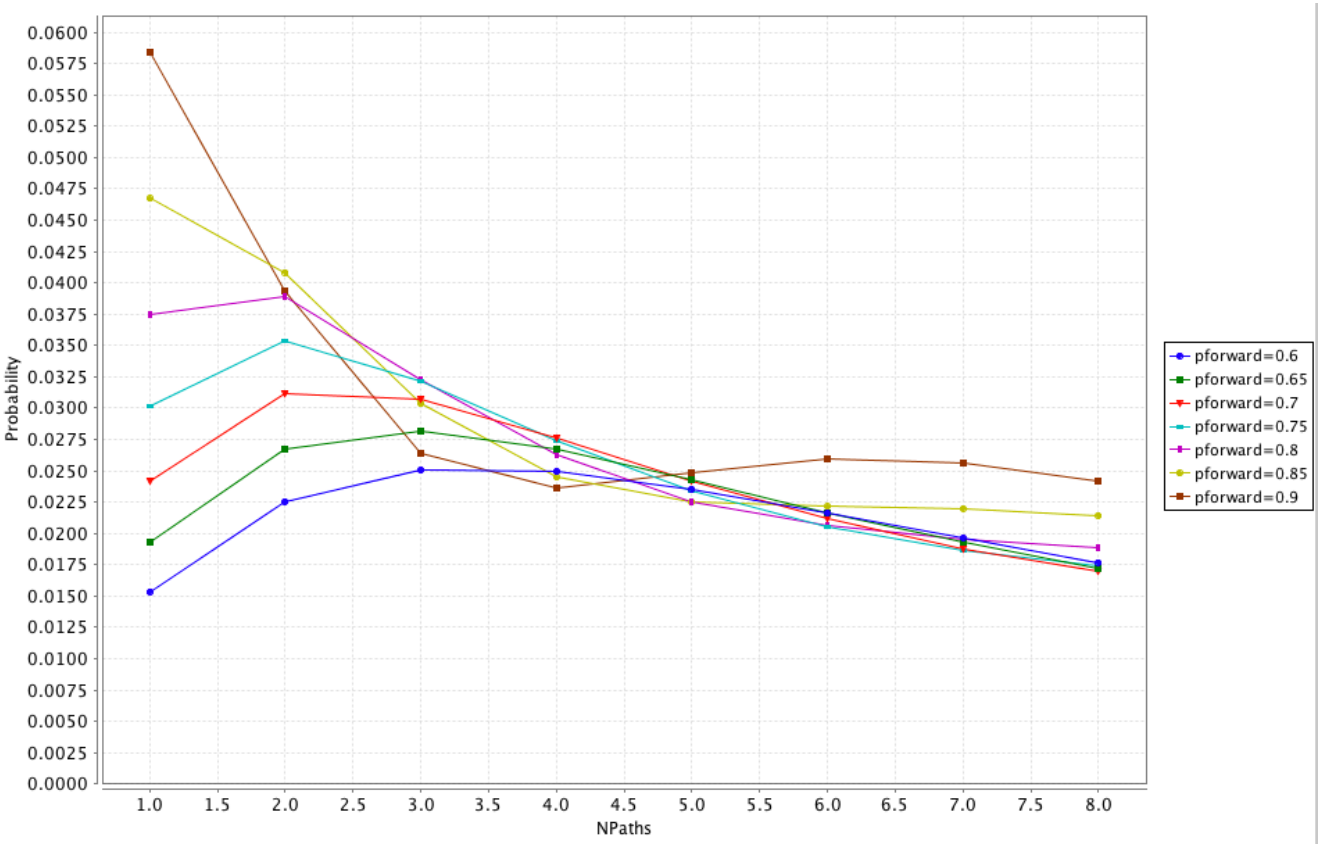
Task 2



For NPaths=1, there is a probability of slightly over pforward x c that the single request will be sent directly to a dishonest part; this is larger for larger pforward. With NPaths=2, there is already a smaller chance that ***both*** requests will be sent directly to a dishonest party, so for a high pforward=0.9 there is a higher chance that one of the two requests will reach a dishonest party through another honest party than the originator. This explains the decrease in the probability of detection. Note that as pforward decreases this is less likely since it is probable that a request will go out directly from the originator or from the next honest party, etc., so the dishonest parties may see a single request, with a greater chance of it coming from the originator.

Task 3

This task is looking at false positives.

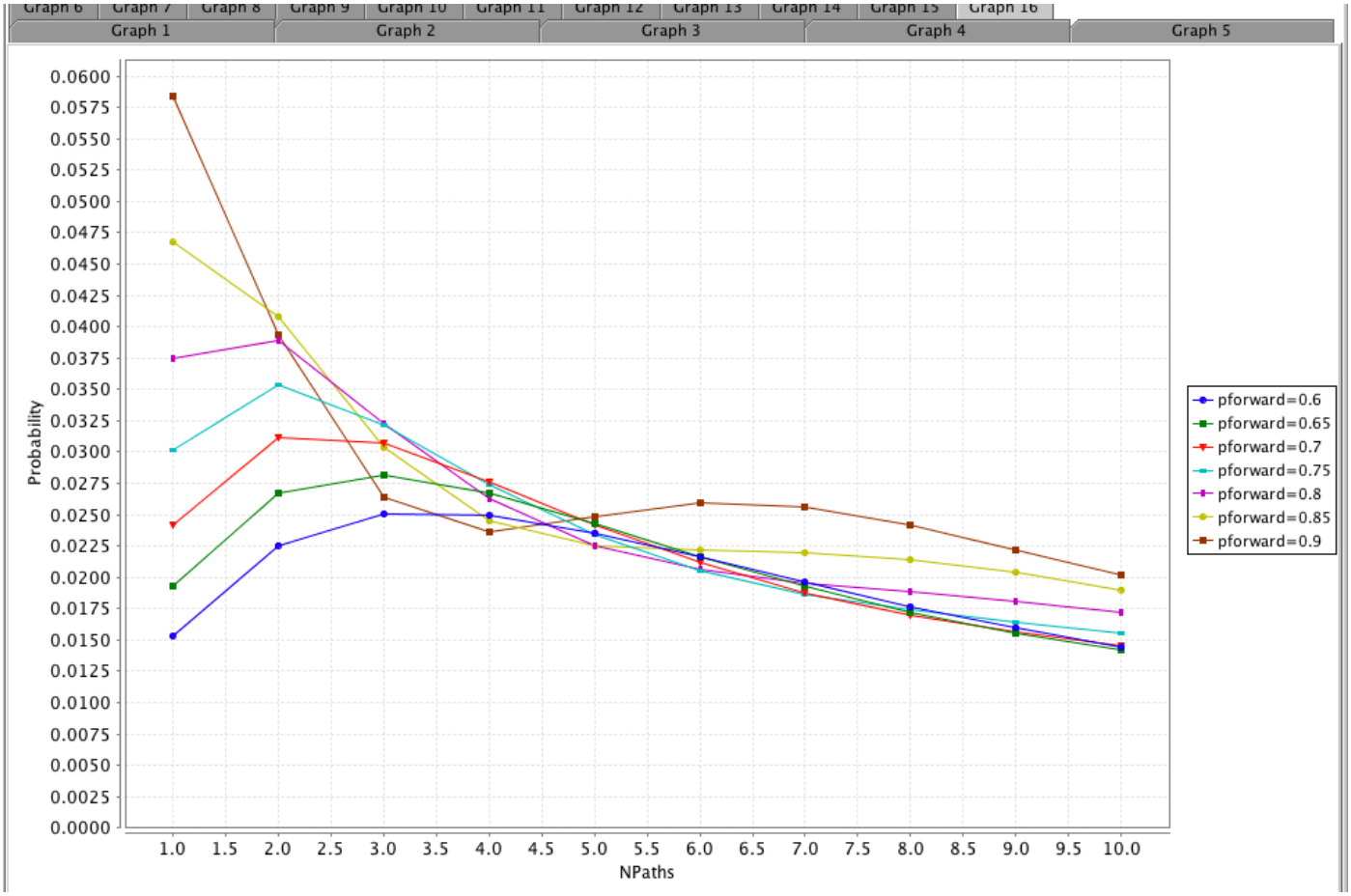


The probability of a false positive is in general very small. For NPaths=1 (a single request), it increases with pforward because for smaller values of pforward there is a lower probability that the request will even reach another party before going out (directly from the originator).

The probability of a false positive tends to ultimately decrease with large numbers of paths, as the actual originator will stand out more and more.

The probability of a false positive for intermediate numbers of paths increases for lower forwarding probabilities because the request/path will leave the group of N parties towards its real destination sooner, so there will be less information for the dishonest users to draw accurate conclusions.

For pforward=0.9 the probability of a false positive has unexpected minima and maxima that show the usefulness of having an automated tool for the analysis. The initial decrease corresponds to more information becoming available to the dishonest parties, but then additional paths temporarily add more noise. The next graph (extended to NPaths=10) show that the overall trend is towards a decrease.



Exercise 3

```
dtmc

const int N;

const double pwrite = 0.001*N*N;
const double pnotice;
const double pguess = (N=4)?0.0001:((N=5)?0.00001:0.000001);
//const double pretry;

module PIN_attack
  state : [0..5] init 0;
  retries : [0..3] init 0; // NEW

  [start]   state=0 -> pwrite:(state'=1) + (1-pwrite):(state'=2);
  [written] state=1 -> pnotice:(state'=3) + (1-pnotice):(state'=2);
  [guess]   state=2 -> pguess:(state'=3) + (1-pguess):(state'=4);
  [success] state=3 -> 1:(state'=3);
  [wrong1]  state=4&retries<3 -> 1:(state'=2)&(retries'=retries+1); // NEW
  [wrong2]  state=4&retries=3 -> 1: (state'=5); // NEW
  [fail]    state=5 -> 1:(state'=5);
endmodule

```

PSEC – Practical 2 Solutions

Verified property: P=?[F state=3]

Lowest probability that PIN will be guessed is for N=4 across the entire range of pnotice values, so it's best to use four-digit PINS for this bank.