Cryptographic Applications of Random Variables

Austin Chase Minor

November 8, 2016

1 Introduction

You are a budding cryptoanalyist and have noticed a security flaw in your companies login system. You find that matching certain keys to values gives you access to peoples account. You have no way of determining which key/value pairs match. Also, you only have so many retry counts. Your job is to determine the number of key/value matches you will get given a certain retry count. This is a mathematical hard problem to determine the exact probability since as a key/value is matched the number of keys decrease and thus the probability increases for a match. So remembering the theory of random variables, you code the problem statement and simulate to find the average number of matches for a given retry count. Below is the mathematical statement of the problem.

2 Mathematical Description

Let A, B be a set of keys and values respectively.
Let $f : A \rightarrow B$ be the function relating keys to values that you are trying to discover.
Let $\phi : A \times B \rightarrow 0, 1$ be a truth function representing $true = 1$ if $f(a) = b; a \in A, b \in B$ and $false = 0$ otherwise.
Let $k \in \mathbb{N}$ represent the number of retry counts allowed (the number of tries for each individual $b \in B$ to the ϕ function.

3 Problem Statement

Through some thought, it can be shown that the best way to go about trying potential values is to try each $b \in B$ with every $a \in Ak$ times removing a as they are matched. This guarantees a minimum of k matches. Furthermore, it maximizes the shared information between $b's$ resulting in a higher probability. The Matlab code implementing this is below.

By repeating this trial over k several times, we can generate a sample of the actual probability. This allows us to estimate the mean and standard deviation along with seeing the general shape of the function. The Matlab code for this along with the corresponding graphs are below.

```
freq = zeros(100, 100);
match_num = 0;

for j = 25:25
  for i = 1:10000
    match_num = match(100,j);
    freq(j, match_num) = freq(j, match_num) + 1;
  end
end

plot(freq(10,:));
```

Listing 1: CryptoSim Main

```
function successes = match(sz, k)
  A = B = 1:sz;
  A = randomize_array(A);
  count = 0;

  for i = 1:sz
    for j = 1:k
      if A(i) == B(j)
        B(j) = [];
        count = count + 1;
        break;
      end
    end
  end
  successes = count;
end
```

Listing 2: Match

```
function B = randomize_array(A)
  sz = size(A);
  sz = sz(2);
  for i = 1:sz
    r = floor(sz * rand(1)) + 1;
    temp = A(i);
    A(i) = A(r);
    A(r) = temp;
  end
  B = A;
end
```

Listing 3: Randomize Array

4 Analysis of Problem