# Project 3

Alex Miller

December 3, 2021

## 1

I generated inputs by picking distinct values to append to *pre* in the search space $S = \{0,1\}^{24}$; $|S| = 2^{24}$. By the birthday principle, with a hash function $H$ with an output space $O$ of 5 bytes, or $2^{40}$ bits, it should only take $q = \sqrt{|O|} = \sqrt{2^{40}} = 2^{20}$ distinct inputs to $H$ to generate a collision. Since $proj3hash$ only generates 40 bits of output, my choice of search space is sufficient to find a collision for problem 1. Furthermore, the probability bound implied by the Birthday principle is consistent with the average number of hashes we generate in our ten trials for problem 1; on average, trials found a collision on $H$ after considering only $2^{20} < 1397456.5 < 2^{21}$ distinct inputs.

## 2

I generated a list of 40 binary transformations (a transformation that is independent of any other in the set) on the two base messages $m_1 =$ "david cash owes alex miller 100 dollars ." and $m_2 =$ "david cash owes alex miller 1,000,000 dollars .". I then took 20 of them and generated all $2^{20}$ possible combinations of transformations on $m_1$ and $m_2$, generating $2^{21}$ distinct inputs for $proj3hash$. It took $2^{20} < 1240967 = 2^{20} + 192391 < 2^{21}$ evaluations to find a collisions between a hash of an $m_1$ based message and an $m_2$ based message.

## 3

I just set *head* to an empty bytearray. On average, my trials needed $2^{20} < 6543315 < 2^{23}$ in order to find a collision on $proj3hash$. The performance of our implementation demonstrates that the bound posited by the theory discussed in class is useful; though our trials needed more than $2^{40/2} = 2^{20}$ evaluations, the observed value was close to the theorized bound.

## 4

I generated a list of 40 binary transformations (a transformation that is independent of any other in the set) on the two base messages $m_1 =$ "david cash owes alex miller 100 dollars ." and $m_2 =$ "david cash owes alex miller 1,000,000 dollars .". I then took 39 of them to be used by $f(x)$, which, after deciding on a base message, applied $transform_i$ to the base message if and only if $x_i = 1$. Therefore, each output of $hash'$ had a distinct mapping in $f(x), \{f(x) : x \in \{0,1\}^{40}\} = 2^{40}$. A collision will not be useful if the distinct inputs $x, x'$ that generated the collision are such that $f(x)$ and $f(x')$ are semantically equivalent. This happens about half the time, assuming $hash'$ is sufficiently uniform. On average, this process should have to run twice in order to generate a useful collision.