Zach Brogan
CS 465
Lab 2

Hash Attack

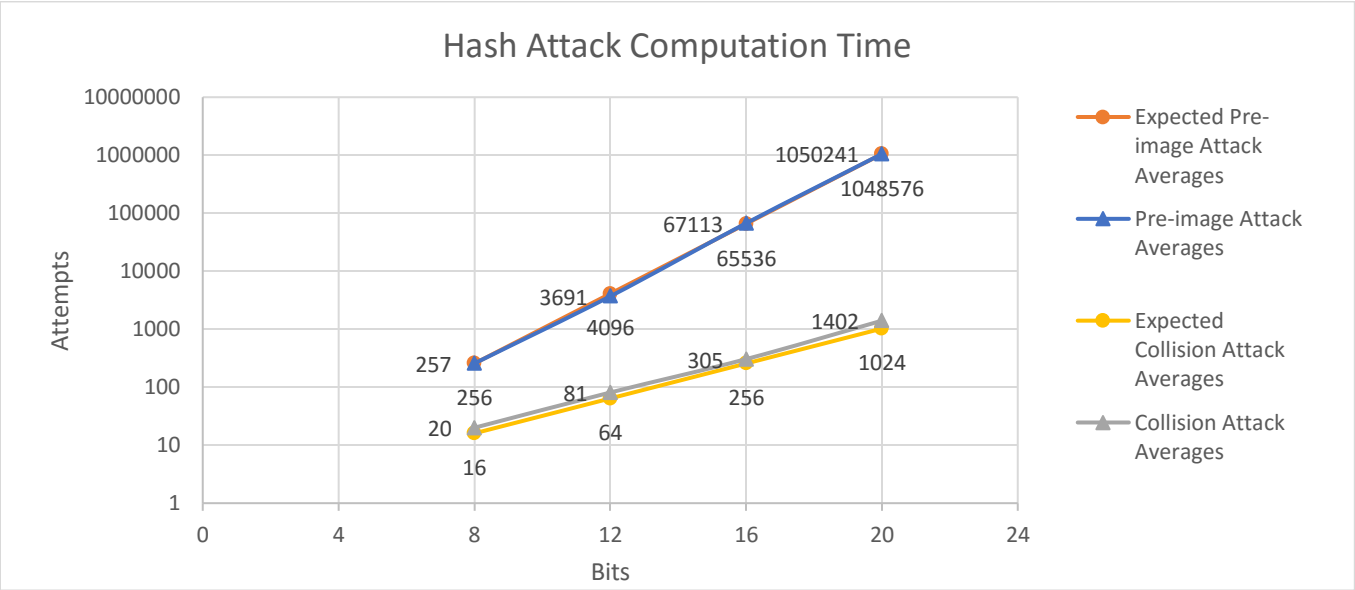
A hash function takes an arbitrarily long input and maps it to an output of a set size. For example, the text "`hello`" maps to "`a1f2fbfe2c4ad81749cd0380b735295d06f9d0c4`" in hex. Ideally, hashes are unique, meaning that it is very hard to find two inputs that produce the same output. This property is referred to as collision resistance. When two inputs to a hash function produce the same digest (output), there is a collision. In general, the greater the digest size, the more collision-resistant the hash function is.

There are two main types of attacks against the collision property of hashes:

1. Collision Attack: Finding any two inputs that map to the same output.
   The expected computation time is `2^(n/2)` calls to the hash function where `n` is the output size in bits.

2. Pre-image Attack: Finding an input that maps to a given output.
   The expected computation time is `2^n` calls to the hash function where `n` is the output size in bits.

I created an experiment in Java to test these ideas. I tested both attacks using the SHA-1 Hash algorithm, truncating the output to 8, 12, 16, and 20 bits and took the average of 50 trials. Below is a graph comparing the computation time of my experimental results to the theoretical. Attempts refers to the number of times the hash function was run with an arbitrary input before a match was found.



As can be seen in the graph, my experiment correlated very closely with the expected results. I only tested up to 20 bits because of the time it would take. It would take hours to compute 50 trials of the 32-bit pre-image attack. The actual 160-bit SHA-1 hash would take even longer (infeasibly long), which is a useful cryptographic property.

Reviewer: Ariel Brogan