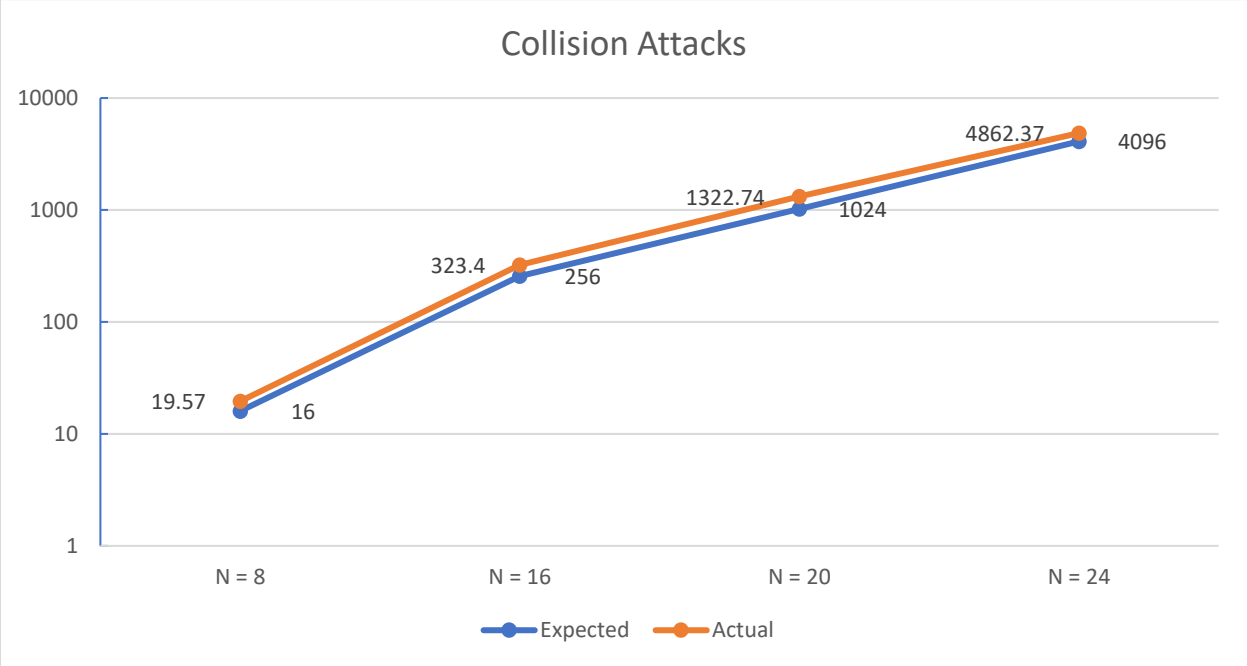
Hector Lopez
Hlj239
Hash Attack

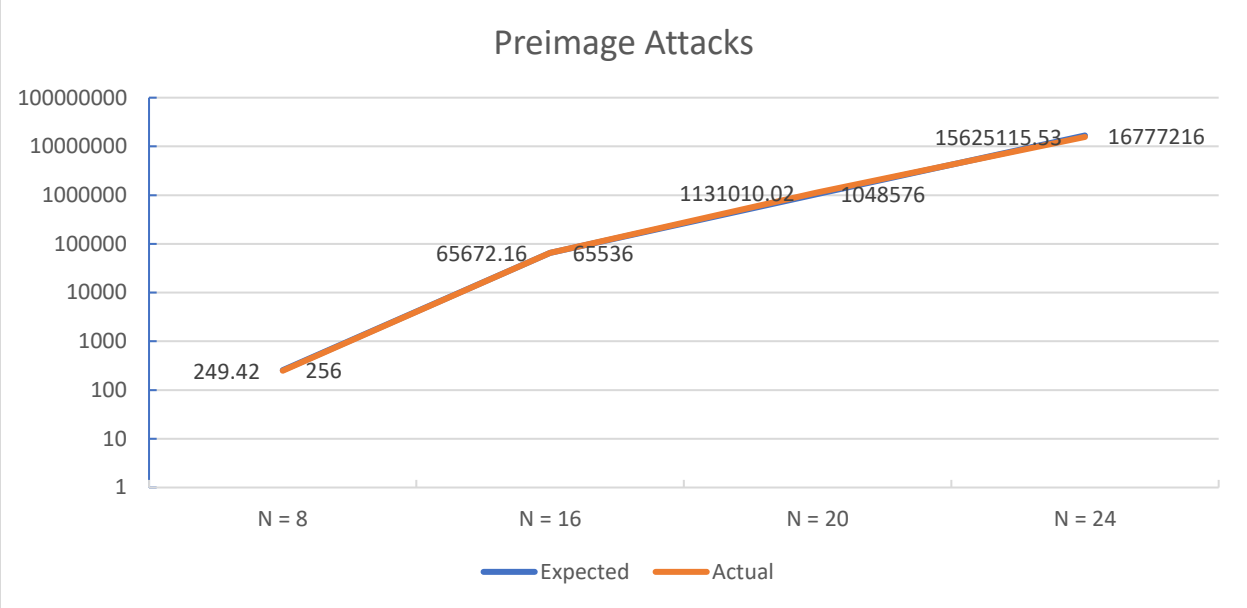
Using the hashlib module, I created a wrapper to create a SHA-1 hash with a given string. The wrapper also took in parameter n, to signify that n most significant bits should be used for the hash.

For the Collision attacks I calculated a hash, set it in to a set and then checked if the new hash is already in the set. If not then I add it and test another hash. The collision attacks have a theoretical 2^(n/2) tries for hashes of n bits. All tests returned higher than the expected average, with the results varying between the actual returning ~18% more tries expected. The variance could be explained by the fact that the Big-O of the theoretical tries of the collision attacks doesn't consider constant factors, which my implementation seems to have. It matches the curve of the expected try, just shifted upwards by about 18%.



For the preimage attacks I first created the hash with a string using n of the most significant bits, then created a random string for a second hash. I then compared the two

strings for equivalence. If they didn't match, I hash the string again and incremented the hex value of the string. This repeats until the strings match. The preimage attacks have a theoretical average of 2^n tries to get an n-bit hash that is equal to the original. For the most part the tests got results that are very close to the average, with most of the actual results in the extremes, 8 and 24 n, being lower. Each n value was tested 300 times. The actual tries came relatively close to the theoretical, with a variance of about 3%



The actual results match the curvature of the theoretical results, and in the case of the preimage attack, they match closely. These attacks however are impractical considering how many tries each one took, and the fact that most SHA-1 hashes have a size n > 24. I tried running n = 32, but after a few hours it still would not finish.

Reviewed By: Andy Bain