Jason Rasmussen
September 12, 2013

Project 2 (Hash Attack) Report

Collision Attack

- **Tests:** To perform a collision attack I generated a set of random bits, truncated the digests, and stored the results in a dictionary until I found two different sets of bits that generated the same digest. I also calculated how many tries it took to find a collision. I ran **20** tests on a bit size of **4, 8, 12, 16, and 24** each, and then took the average number of tries among the 20 tests.

- **Results:** (Bits, Average tries for 20 tests)
    - 28, 21248
    - 24, 4906
    - 20, 1396
    - 16, 340
    - 12, 63
    - 8, 16

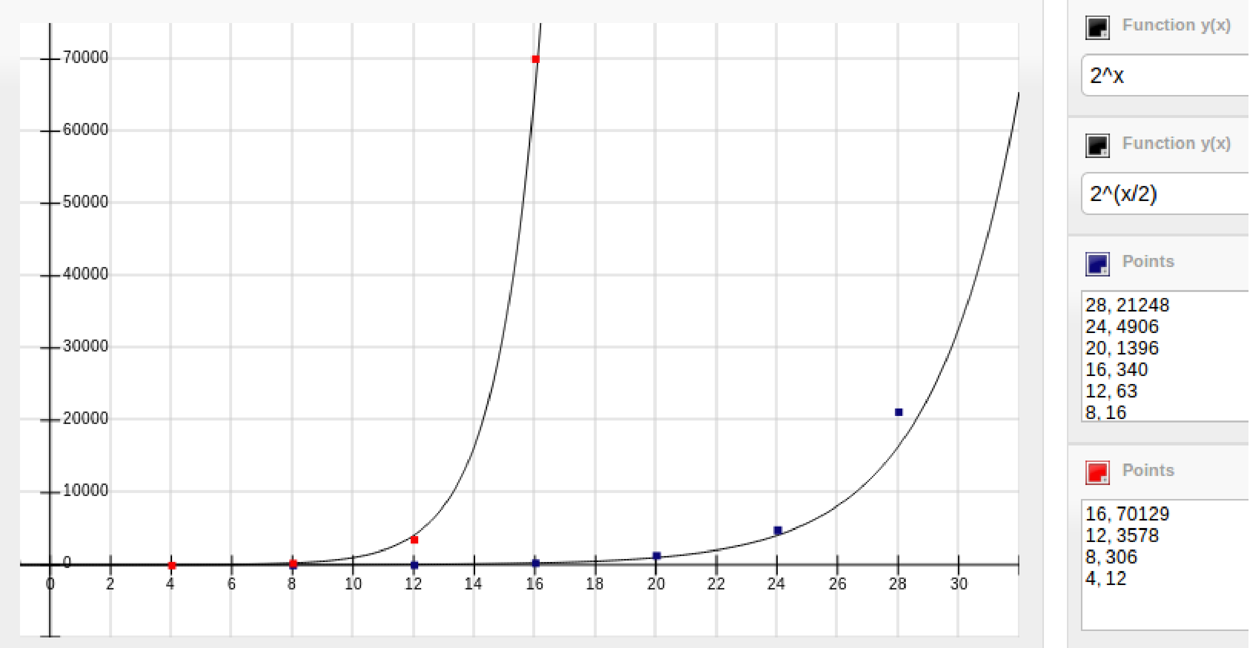
- **Summary:** In conclusion, my empirical data shows that, on average, a collision attack takes 2^(n/2) tries, where n is the number of bits in the digest. This is apparent in the different bit sizes and is shown in the figure at the bottom of the report.

Preimage Attack

- **Tests:** To perform a preimage attack I first generated a set of random bits and stored its digest. I then proceeded to generate additional sets of random bits and compared the resulting digests against the original digest. I did this until I found a second set of bits that generated the original digest. I also calculated how many tries it took to find this second set of bits. I ran **20** tests on a bit size of **4, 8, 12, and 16** each, and then took the average number of tries among the 20 tests.

- **Results:** (Bits, Average tries for 20 tests)
    - 16, 70129
    - 12, 3578
    - 8, 306
    - 4, 12

- **Summary:** In conclusion, my empirical data shows that, on average, a preimage attack takes 2^n tries, where n is the number of bits in the digest. This is apparent in the different bit sizes and is shown in the figure below.



Function y(x)
2^x

Function y(x)
2^(x/2)

Points
28, 21248
24, 4906
20, 1396
16, 340
12, 63
8, 16

Points
16, 70129
12, 3578
8, 306
4, 12

The first line represents y = 2^n, while the second line represents y = 2^(n/2). The red dots (listed to the bottom right) are the results of the preimage test, while the blue dots (listed to the top right) are the results of the collision test.

Implementation Notes:

- I used Python’s os.urandom() method to generate random bits
- I used Python’s hashlib.sha1() method to generate a hex digest