Assignment: Cryptographic Hash Functions

Caleb Dennis

2/7/22

CPE321-03

## Task 1: Exploring pseudo randomness and collision resistance

Using the pycrypto library it is easy to hash strings. By printing these strings to the outputs as byte arrays I observed that not only do strings with substantially different content result in dissimilar hashes, but strings with nearly identical content also result in dissimilar hashes. Below is the console output with my results.

hashes of some strings

b'\x81\x140\x90n\x0f\x8f\xd9\xab\x92\x9b\xa2mi9s\xdf\xf0\xf9vL\x17\xa3\xc4\xdf\xe2j4\xb4\x8f\x02\xa1'

b'f\t#\xa7C\x8d\xee\xd7\x7f1\xb4\xa2m|\x94O1QL\xd4\xd6\x1f\xef\xc5\xc8Nc?\xbd C\xc1'

hashes of strings which differ by a single bit

b'\x8c\xef\xcd\xfaf\xac"\xff\xbal\x80Y\x7fO\x94\x07w\xaf\xb90\'pG\xab:R\xf2\xdb\xb83i8'

b'q\xb7\x80e\x00.\x98\xf3Q\x85g\x92\xc3b1g\xa1P=\x1c\xb2;\xc5\x91^B\xf4m\x8d]D\xb6'

b'\xc4\x08\xf6\x9e\x8b\x8f\xf3\xf2\xd3i\xc3J\x9d\t\xaaO0l\xe7\x91 \x06\xaa\xccj\xe6\x87\x8c\xb9\xd0ke'

b'Mp\x86\xd9@(\xadB%\x16\xbe\xd4\xe2\xd8\x13\xe1\xee\xf40\x8d\x02\xfe\xc9\xad\xed\xd2FWz\xcb\xfe\xc9'

b'\xfc\xbb\xb0}\xe0\x96\xb4}\xa9\xb17>F\x0fU\x7f/\x93n\xf1u\x14\x88N\xb1w\x9dB\xbc5&\x05'

Chart, scatter chart

Description automatically generatedWhen searching for hash collisions, the amount of work to find a collision increases exponentially with the number of bits in the hash. Finding a collision on an 8 bit hash took only a few milliseconds, finding collisions on a 50 bit hash took nearly an hour. I also got lucky with the 48 bit hash and found a collision much earlier than expected.

A picture containing chart

Description automatically generated

## Task 2: Breaking real hashes