Programming Assignment 2
Brandon Lee
CS 370 - Bobba

Task 1 - One Way Hash and Their Randomness

Generate the hash value H1 for this file using SHA1 (SHA256). (Add this value in your writeup)
H1: SHA1(sample.txt)= 3fb6542ba0cc9b35cf14bd8db9d33ac9669a898f
H1: SHA256(sample.txt)=
9ed7b218f3402376f0d2f77b89093704515f68f82857f17c9ba8dc4348ecf81f

Flip one bit of the input file. You can achieve this modification using hex editors like ghex or Bless.

I flipped the last bit from a 0 to 1:
(01010100011010000110010100100000011000110110111101110111001000000110101001110101011011010111000001110011001000000110111101110110011001010111001000100000011101000011010000110010100100000011011010110111101110111011011**1**)

Generate the hash value H2 for the modified file.(Add this value in your writeup)

H2: SHA1(sample.txt)= a8bcd1b7f07e1c7a8d9fea63b7439a9e82380c4a
H2: SHA256(sample.txt)=
2a5fa108e172dbfa68ed9a0f7307539f7ef07bbbab58ea01ec2b3f9e5ea13ab3

Please observe whether H1 and H2 are similar or not. How many bits are the same between H1 and H2.

H1 SHA1 Binary:
00111111101101100101010000101011101000001100110010011011001101011100111100010100101111011000110110111001110100110011101011001001011001101001101010001001100001111

H2 SHA1 Binary:
10101000101111001101000110110111111100000111111000011100011110101000110110011111111101010011000111011011101000011100110101001111010000010001110000000011000100101010

88 similar bits

H1 SHA256 Binary:
10011110110101111011001000011000111100110100000000100011011101101111000011010010111101110111101110001001000010010011011100000100010100010101111101101000

11111000001010000101011111110001011111001001101110101000110111000100001101001000111011001111100000011111

H2 SHA256 Binary:
00101010010111111010000100001000111000010111001011011011111110100110100011101101100110100000111101110011000000111010100111001111101111110111100000111101110101010110101100011101010000000011110110000101011001111111001111001011110101000010011101010110011

130 similar bits

Flip bits 1, 49, 73, and 113 and record the number of bits that are different between H1 and H2 when using SHA1 and SHA256 in a table. What trend do you see in table? Does this trend change if you flip different bits in the file or flip multiple bits?

H1 Flipped Binary (1, 49, 73, 113):
00010100011010000110010100100000011000110110111100110111001000000110101000110101011011010111000001110011001000000010111101110110011001010111001000100000011101000011010000110010100100000011011010110111101101111011011100010111000101110

Text: he co7 j5mps /ver the moon.
SHA1(sample.txt)= aa4ab6e61b5ba4b0aba619048435c35d19caed3e
SHA256(sample.txt)=
745c3bf420ff59f221ac17ea4bc0a36d33ee89b54dbdb57e387b5b2839b58ef1

H2 Flipped Binary (1, 49, 73, 113):
00010100011010000110010100100000011000110110111100110111001000000110101000110101011011010111000001110011001000000010111101110110011001010111001000100010011101000011010000110010100100000011011010110111101101111011011100010111000101110

Text: he co7 j5mps /ver"the moon.
SHA1(sample.txt)= 6a1f2314b08fe55cac79c21c5e4b7671414ff31f
SHA256(sample.txt)=
eae09128f3b937c3023c1f753223d6f0c53a9bed1a0eb1d6189a86d9af7f5c35

SHA1 H1 vs H2:
10101010010010101011011011100110000110110101101110100100101100001010101110100110000110010000010010000100001101011100001101011101000110011100101011101101000111110

01101010000111110010001100010100101100001000111111100101010111001010110001111001110000100001110001011110010010110111011001110001010000010100111111111001100011111

80 similar bits

SHA256 H1 vs H2:
0111010001011100001110111111010000100000111111110101100111110010001000011010
11000001011111101010010010111100000001010001101101101001100111110111010001001
1011010101001101101111011011010101111110001110000111101101011011001010000011
10011011010110001110111110001

1110101011100000100100010010100011110011101110010011011111000011000000100011
1100000111110111010100110010001000111101011011110000110001010011101010011011
1110110100011010000011101011000111010110000110001001101010000110110110011010
1111011111101011100001 10101

129 similar bits

Discussion:
In the above results, bits 1, 49, 73, and 113 were flipped in both H1 and H2. Subsequently, both H1 and H2 were both hashed with SHA1 and SHA256. I added an extra bit flip to H2. Because both hashes have essentially the same bits flipped, the similar bit count appears to come close to their non-flipped counterparts with talleys of 80 similar bits and 129 similar bits for SHA1 and SHA256 respectively.

Task 2 - Keyed Hash and HMAC

Keys are just varying lengths of ‘aaa...a’

128 bits
HMAC-SHA1(sample.txt)= c31b45626f2f02d4d26784ee33f9ffbcba91e908
HMAC-SHA256(sample.txt)=
750cac5e7a5018d0886ed640f6007f763929d0a8a0be04900c71e16e75280e1b

160 bits
HMAC-SHA1(sample.txt)= 55eac3f8afa436fc1a9d1b610fcfe77188c971c2
HMAC-SHA256(sample.txt)=
3791c3cb16d7cffd5f092943ba551c58665614835b8bb5a0fde5a5eaa5348ce1

256 bits
HMAC-SHA1(sample.txt)= 4902f84d247445dafd7037d432c4882c7d4beeb4
HMAC-SHA256(sample.txt)=
a242f9f4a91988dbaab21857ca05801bcc45baffdcd41b793b4784fb126bb8f7

Do we have to use a key with a fixed size in HMAC? If so, what is the key size? If not, why?

Yes. The key should be the same length as the hash output. For example, if we are using SHA256, we would need a 256 bit key in order to guarantee integrity. Otherwise, the security argument of HMAC would not hold.

http://security.stackexchange.com/questions/45523/what-key-to-use-in-hmac
http://crypto.stackexchange.com/questions/15551/why-does-hmac-need-a-fixed-length-padding

Task 3 - Weak versus Strong Collision Resistance Property

Tabular Data

	Weak	Strong
1	5583576	9975198
2	74775947	10451677
3	1683383	47049185
4	2852838	4642141
5	11684141	4521348
6	4545580	81954391
7	13470046	64056145
8	1409121	8104087
9	1409121	36178353
10	4894647	33269842
11	20017895	25398617
12	33001480	13528128
13	34398413	15564308
14	10822842	66502518
15	8630048	11274135
16	10412895	31549528
17	4821724	10471561
18	7931209	9466076
19	14299087	9059036
20	8146731	4198425
21	13168336	10623401
22	13974925	15249158

23	6619413	1272987
24	6006780	34072593
25	35757179	8745934
26	11729165	3212906
27	9694989	782744
28	831152	26463652
29	38369637	15669363
30	38366582	2339096
31	19622289	31094672
32	23983741	22660558
33	23693047	25933582
34	12936852	9209756
35	13328916	12154984
36	14533074	4204273
37	4503373	13951027
38	25627307	13555051
39	378412	11686379
40	378412	7296707
41	378412	24370516
42	378412	30554285
43	5359357	6374662
44	21958081	1024924
45	17941534	16138447
46	19406935	42195323
47	8478260	30527962
48	113788	49025619
49	113788	2288802
50	113788	21734114
51	113788	2856383
52	113788	37954290
53	113788	4118542
54	113788	6208432
55	113788	11244252
56	113788	55839631

57	13125268	25816487
58	2273320	25875308
59	6960992	57965473
60	18782014	1675454
61	21260302	8805455
62	8510837	8295270
63	2180113	10709873
64	59218159	24416393
65	3446091	1497663
66	3011435	1863548
67	42704340	59798207
68	21368367	20804315
69	7243484	3005888
70	16553545	4359944
71	9767462	11764472
72	56413421	423182
73	2474785	423182
74	18054119	12791315
75	27498945	18582349
76	26971566	38557060
77	7632555	4048664
78	5623661	11746602
79	17116564	6765590
80	4901197	7234762
81	56965606	14870510
82	4412869	11899182
83	26818976	1807209
84	886521	7693583
85	13066930	15067780
86	11669802	7859419
87	722179	5773147
88	722179	10830637
89	17075450	3390794
90	10399667	9030015

91	11994566	19877223
92	2574919	4681133
93	10241344	24763957
94	3428626	7927262
95	3641818	6925301
96	5965680	21294346
97	2544258	5693515
98	9514152	4057250
99	25737209	32694032
100	4750161	2907785

How many trials will it take you to break the weak collision resistance property using the brute-force method? You should repeat your experiment multiple times (100 or more depending on how long each trial takes), and tabulate your findings on number of trials in each experiment along with the statistics (average, median).

Average **12894448.42**

Median **8570442.5**

How many trials will it take you to break the collision-free property using the brute-force method? Similarly, to the above task repeat the experiment multiple times and and tabulate your findings on number of trials in each experiment along with the statistics (average, median).

Average **16961202.42**

Median **11037444.5**

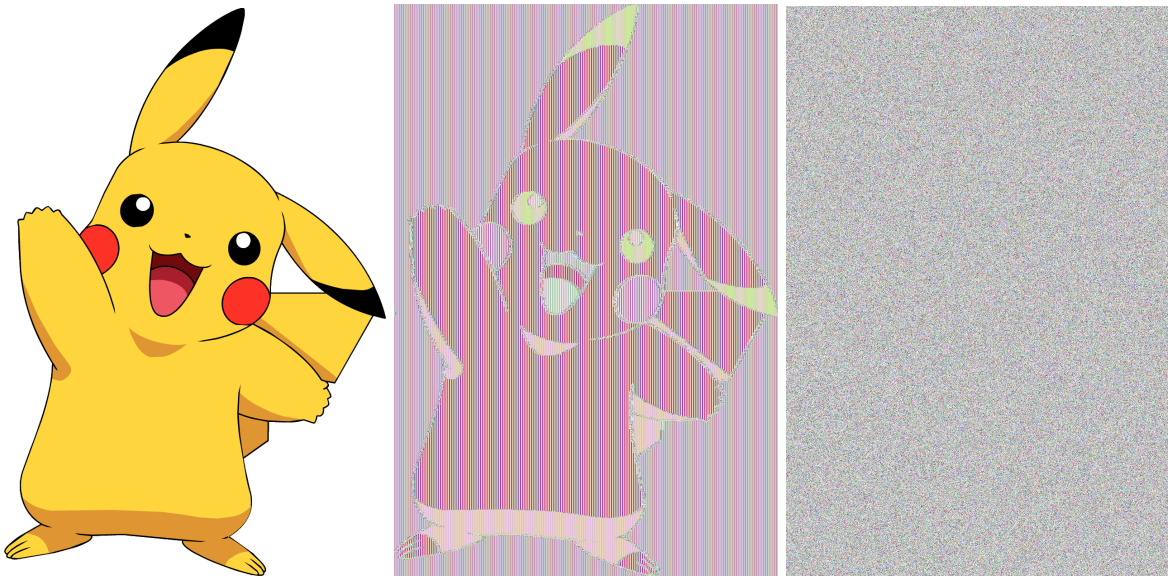
Based on your observation, which property is easier to break using the brute-force method? Can you explain the difference in your observation mathematically?

Purely based on my observation above, weak collision resistance is easier to break, with an average of 12894448.42 (weak) to 16961202.42 (strong). I believe that this is due to my methodology behind performing both. For weak collision resistance I hashed a “known” value to

compare to and generated random values until a match was found. For strong collision resistance I simply generated two random values each iteration and compared them until a match was found. Mathematically in regard to each of these processes, weak collision resistance had a better chance at finding a match in $H(a) = H(b)$, and $a \neq b$.

However in terms of brute force effectiveness in weak versus strong collision resistance overall, I'm not quite sure how to explain my results to the comparison between weak and strong collision resistance above, as I believe that had the strong values been stored into some sort of data structure and the collective data been compared to a generated value, strong collision resistance would've yielded much more brute force breakable results.

Task 4 - Encryption using different ciphers and modes



Original | ECB | CBC

Task 5 - Encrypting with OpenSSL

Please refer to part5/ directory for code.