Assignment 4 - Lab Report
Student: Shuo Yang
NetID: shuoyang@email.arizona.edu
GradeID: 50

**Task-1**

```
[10/12/2015 22:47] seed@ubuntu:~/crypto_hashing$ openssl dgst -md5 inputfile
MD5(inputfile)= ed076287532e86365e841e92bfc50d8c
[10/12/2015 22:47] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha1 inputfile
SHA1(inputfile)= 2ef7bde608ce5404e97d5f042f95f89f1c232871
[10/12/2015 22:47] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha inputfile
SHA(inputfile)= 1261178ff9a732aacfece0d8b8bd113255a57960
[10/12/2015 22:50] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha256 inputfile
SHA256(inputfile)= 7f83b1657ff1fc53b92dc18148a1d65dfc2d4b1fa3d677284addd200126d9069
[10/12/2015 22:47] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha512 inputfile
SHA512(inputfile)=
861844d6704e8573fec34d967e20bcfef3d424cf48be04e6dc08f2bd58c729743371015ead891cc3cf1
c9d34b49264b510751b1ff9e537937bc46b5d6ff4ecc8
```

The hashing value produced md5 algorithm tends to be short (128 bits). Both sha and sha1 produce 160 bits hashing value. Sha256 and sha512 produce 256 bits and 512 bits hashing value correspondingly.

**Task-2**

```
[10/12/2015 23:05] seed@ubuntu:~/crypto_hashing$ openssl dgst -md5 -hmac "abcdefg"
inputfile
HMAC-MD5(inputfile)= 20be7a41e26808c7aad8a07284d0af13
[10/12/2015 23:25] seed@ubuntu:~/crypto_hashing$ openssl dgst -md5 -hmac
"abcdefghijkmnopq" inputfile
HMAC-MD5(inputfile)= 68b7999ba9a4cad128a329e16b871cdf
[10/12/2015 23:26] seed@ubuntu:~/crypto_hashing$ openssl dgst -md5 -hmac "123"
inputfile
HMAC-MD5(inputfile)= 91b2e463b5b2fd6c389834a8dee5630f
[10/12/2015 23:26] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha1 -hmac "abcdefg"
inputfile
HMAC-SHA1(inputfile)= 1feee7729f77c05b3abf2977a5645480cc92aeab
[10/12/2015 23:26] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha1 -hmac
"abcdefghijkmnopq" inputfile
HMAC-SHA1(inputfile)= 099e30e49825711d765f92570b1a5e0d1bf8afeb
[10/12/2015 23:26] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha1 -hmac "123"
inputfile
HMAC-SHA1(inputfile)= 5471001127ed5f6261576c66033d9cbe37c7c91e
[10/12/2015 23:26] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha256 -hmac
"abcdefg" inputfile
HMAC-SHA256(inputfile)=
81690fdf3e5bac19f3c304c9807dced229b14777d495ca5cf0ffb484d5ee6bd8
[10/12/2015 23:27] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha256 -hmac
"abcdefghijkmnopq" inputfile
HMAC-SHA256(inputfile)=
```

```
7dade113e13aaaeb8f58a3b13364250b52d920c6bb30f17046ebc881ff19f11a
[10/12/2015 23:27] seed@ubuntu:~/crypto_hashing$ openssl dgst -sha256 -hmac "123"
inputfile
HMAC-SHA256(inputfile)=
28240b6106d7b6f5cd703616f76f0b44a91fcc98b6b96869293391d7fa8340ff
```

We don't have to use a key with fixed size, because fix sized key is easy to be recovered by using brute-force method. Variable key length makes such attack infeasible.

**Task 3**

input file: “Hello World!”
input file being flipped: “Hdllo World!”

Using md5:

```
H1 = ed076287532e86365e841e92bfc50d8c
H2 = d47b1bc0e58f62b44b678b8b88ab411b
```

```
H1 xor H2 = 0x397c7947b6a1e48215e39519376e4c97L
```

H1 and H2 are significantly different. They have 62 bits in common out of the total 128 bits.

Using sha256:

```
H1 = 7f83b1657ff1fc53b92dc18148a1d65dfc2d4b1fa3d677284addd200126d9069
H2 = 36ec6e8c81b9de3ca62c1bfcd94e454c383e46853b43cecd24e1f2f5deae7e12
```

```
H1 xor H2 = 0x496fdfe9fe48226f1f01da7d91ef9311c4130d9a9895b9e56e3c20f5ccc3ee7bL
```

H1 and H2 are significantly different. They have 119 bits in common out of the total 256 bits.

**Task 4**

4.1) One-way property of hash functions states that it is practically impossible (in terms of computational complexity) to recover the message from its hashing value alone.

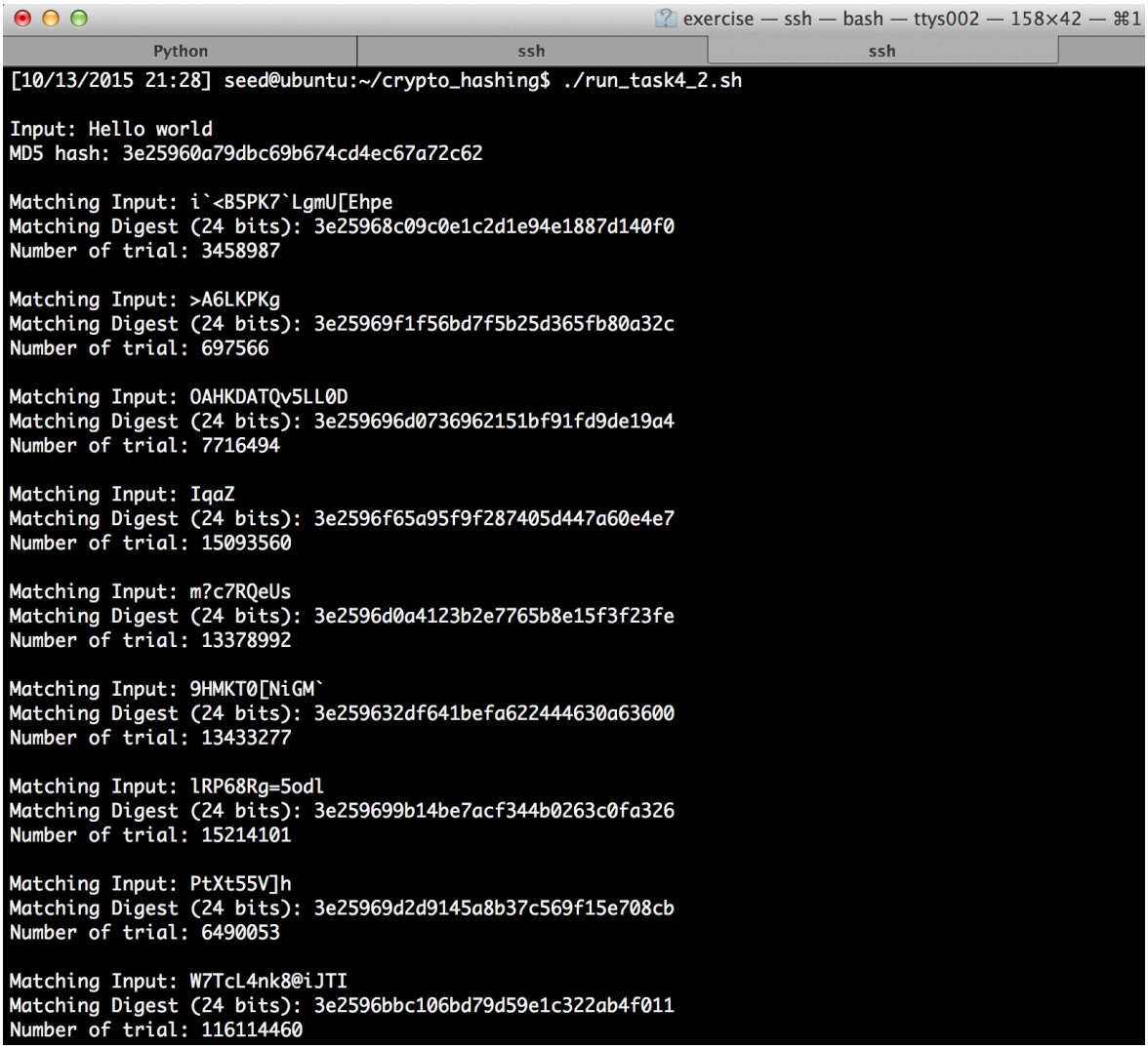
Collision-free property states that given a message *M* and its hash value *H(M)*, it is very hard to find a different message *M'* such that *H(M) = H(M')*.

4.2) I ran the code I wrote for this task for 10 times and collect the average number of trials needed for breaking the one-way property.

Average number of trials: **21894432**

| Run #1 | Run #2 | Run #3 | Run #4 | Run #5 | Run #6 | Run #7 | Run #8 | Run #9 | Run #10 | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| 3458987 | 697566 | 7716494 | 15093560 | 13378992 | 13433277 | 15214101 | 6490053 | 116114460 | 27346835 | **21894432** |

Below is the screenshot:

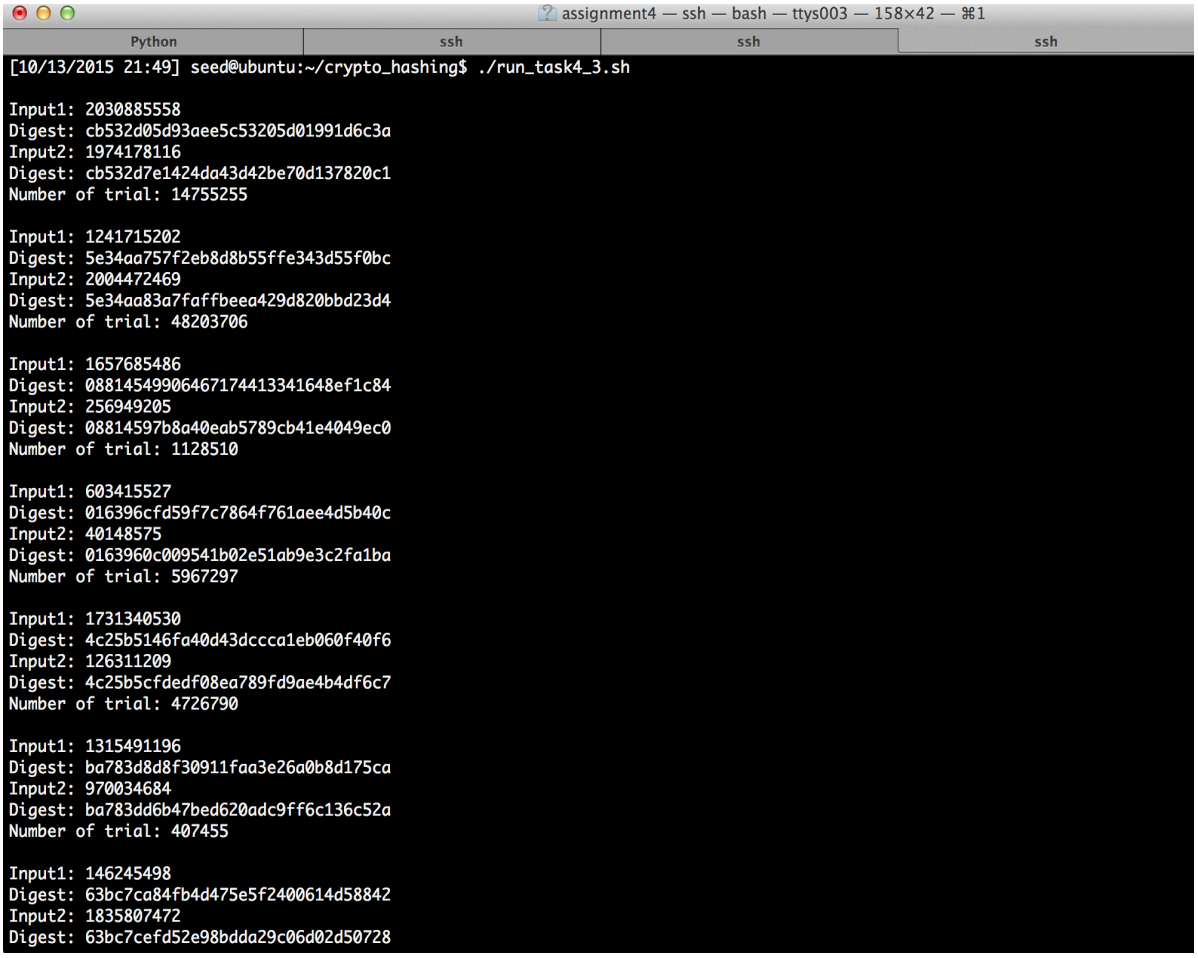



4.3) I ran the code I wrote for this task for 10 times and collect the average number of trials needed for breaking the collision-free property.

Average number of trials: **12265037**

| Run #1 | Run #2 | Run #3 | Run #4 | Run #5 | Run #6 | Run #7 | Run #8 | Run #9 | Run #10 | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| 14755255 | 48203706 | 1128510 | 5967297 | 4726790 | 407455 | 10414732 | 3025284 | 21687897 | 12333448 | **12265037** |

Below is the screenshot:




4.4) Since the average number of trials for breaking the one-way property is **21894432**
and the average number of trials for breaking the collision-free property is **12265037**, it seems that collision-free property is easier to break.