Self Study Lab

Lab: Cryptographic Hash

0. Objectives
To learn to use message digests and secure hash functions.
Library use in this lab: OpenSSL as an open source implementation of SSL and TLS protocols

1. Introduction
The functions of hashing are one-way functions which make it possible to calculate a check code (print) related to the message to be transmitted. It is impossible (or at least very difficult) to modify the message to obtain the same print.

In this lab, we will play with various one-way hash algorithms. You can use the openssl command to generate the hash value for a file. To see the manuals, you can type

```
$ man openssl
```

2. Lab Tasks 1: Get familiar with hashing using `openssl`
2.1 Short guide of `openssl` for hash generations

You can use `openssl` to generate hash in several ways. The typical command can be:
```
openssl [dgsttype] [filename]
openssl dgst -[dgsttype] [filename]
```

You can list all digest using the following commands:
```
$ openssl dgst -h
$ openssl list-message-digest-commands
```

The examples below show how to generate hash with md5 algorithm from the file `srcfile`.
```
$ ls -l >srcfile
```
```
$ openssl dgst -md5 srcfile
```
```
$ openssl dgst -sha srcfile
```

You can also generate a hash from a phrase liked:
```
$ echo -n “Text2Hash” | openssl dgst -md5
$ echo -n “Text2hash” | openssl dgst -sha1
```

Use the following commands to manually create a file and find the corresponded sha1 hash
```
$ echo -n "Password" |openssl dgst -sha1 > hash1.txt
$ echo -n "password" |openssl sha1 >> hash1.txt
$ echo -n "Admin123" |openssl sha1 >> hash1.txt
$ echo -n "QUERTYUIO" |openssl sha1 >> hash1.txt
$ more hash.txt
```

If you already have a list of words then the following bash script can be used to automate the SHA1 generation, reading each line in a file, then generating a file off the resulting hashes. Replace 'wordlist' with the file path of your word list.

For the following example, create several of desired words in the file `wordlist` and then execute the following command.

```
$ for i in $(cat wordlist); do echo -n "$i"| openssl dgst -sha1 >> hash2.txt done
```

2.2 Hash performance

Using `openssl speed [digest_type]` to compare hash performance of **md4, md5, ripemd160, sha1, sha256** and **sha512** in term of block size throughput. Analyze the results and write your analysis of the results.

Give an analysis about performance results:

```
[lab7@lab7 lab3]$ sh speedtest.sh
Doing md4 for 3s on 16 size blocks: 10380636 md4's in 3.00s
Doing md4 for 3s on 64 size blocks: 10650540 md4's in 2.99s
Doing md4 for 3s on 256 size blocks: 5922834 md4's in 3.00s
Doing md4 for 3s on 1024 size blocks: 2757290 md4's in 2.99s
Doing md4 for 3s on 8192 size blocks: 414813 md4's in 3.00s

The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
md4               55363.39k   227971.42k   505415.17k   944302.66k  1132716.03k

Doing md5 for 3s on 16 size blocks: 10964612 md5's in 3.00s
Doing md5 for 3s on 64 size blocks: 7584770 md5's in 2.99s
Doing md5 for 3s on 256 size blocks: 4471814 md5's in 3.00s
Doing md5 for 3s on 1024 size blocks: 1884236 md5's in 2.99s
Doing md5 for 3s on 8192 size blocks: 278099 md5's in 3.00s

type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
md5               58477.93k   162349.59k   381594.79k   645303.57k   759395.67k

Doing rmd160 for 3s on 16 size blocks: 7692699 rmd160's in 2.99s
Doing rmd160 for 3s on 64 size blocks: 4880112 rmd160's in 3.00s
Doing rmd160 for 3s on 256 size blocks: 2332120 rmd160's in 2.99s
Doing rmd160 for 3s on 1024 size blocks: 751695 rmd160's in 3.00s
Doing rmd160 for 3s on 8192 size blocks: 102155 rmd160's in 3.00s

type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
rmd160            41164.94k   104109.06k   199673.15k   256578.56k   278951.25k

Doing sha1 for 3s on 16 size blocks: 11627249 sha1's in 2.99s
Doing sha1 for 3s on 64 size blocks: 8431130 sha1's in 3.00s
Doing sha1 for 3s on 256 size blocks: 4825190 sha1's in 2.99s
Doing sha1 for 3s on 1024 size blocks: 1763020 sha1's in 3.00s
Doing sha1 for 3s on 8192 size blocks: 254727 sha1's in 2.99s

The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
sha1              62219.39k   179864.11k   413126.64k   601777.49k   697900.86k

Doing sha256 for 3s on 16 size blocks: 11099285 sha256's in 2.99s
Doing sha256 for 3s on 64 size blocks: 6202413 sha256's in 3.00s
Doing sha256 for 3s on 256 size blocks: 2678986 sha256's in 3.00s
Doing sha256 for 3s on 1024 size blocks: 822909 sha256's in 2.99s
Doing sha256 for 3s on 8192 size blocks: 110575 sha256's in 3.00s

type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
sha256            59394.17k   132318.14k   228606.81k   281825.69k   301943.47k

Doing sha512 for 3s on 16 size blocks: 7558522 sha512's in 2.99s
Doing sha512 for 3s on 64 size blocks: 7617853 sha512's in 3.00s
Doing sha512 for 3s on 256 size blocks: 2863959 sha512's in 3.00s
Doing sha512 for 3s on 1024 size blocks: 999929 sha512's in 3.00s
Doing sha512 for 3s on 8192 size blocks: 140430 sha512's in 2.99s

type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
sha512            40446.94k   162514.20k   244391.17k   341309.10k   384750.02k


size 16: sha1 ->; md5 ->; sha256 ->; md4 ->; ripemd160 ->; sha512
size 8192: md4 ->; md5 ->; sha1 ->; sha512 ->; sha256 ->; ripemd160
```

2.3 Hash properties
Do a research from the net and fill the table. Hello world my name is ...

Algorithm	Output size (bits)	Block size (bits)	Sample Hex Output
md4	128	512	c499858f9346321f134008b9981b780c
md5	128	512	557bd0e129829b901050eef7d851cea8
ripemd160	160	512	3513f47d40b802d5a2a0b97b08b5800c3657f989
sha1	160	512	48a3e441f2649976c0a823e62ecdcdd1b4b4dd40
sha224	224	512	97c75b741b7b4404224cfdd97e7cfca121ad7ac053c7d4f4bf556621
sha256	256	512	27ed1c28ad0ab7bafd9faae2f29b81d1a226bb85fc13f5aaa82c471eda9b9de2
sha384	384	512	e6548d838cedd45d68c3dcf77ff8c7fb8d8a91c7028452b152b2e8f5260c4dae6989668d9c7e0a035a66a64fd78cf908
sha512	512	576	c1dd65c97ee92bec4ce9906c982bb3036964998d298d6e57b80ad2e1b51d9b128c381360879b7b500c9d0d9789595194c72c807434259bac3d8a49c29b53a581
streebog256	256	512	`0x 3f539a213e97c802cc229d474c6aa32a825a360b2a933a949fd925208d9ce1bb`

streebog512	512	512	`0x d2b793a0bb6cb5904828b5b6dcfb443bb8f33efc06ad09368878ae4cdc8245b9 \`
			
			`7e60802469bed1e7c21a64ff0b179a6a1e0bb74d92965450a0adab69162c00fe`

whirlpool	512	512	eaea438cfc9ad2b2e209f1553ec7d187f3482d4098ad639bd2c6fd4ed4b164a554e3c4a64b4e5e50f1eaf222f65fb61add46395bd0171db0aa4dd5f99b5706f6

3. Hash collisions Exercises (Folder `collision_ex`)
3.1 Investigation of MD5 Collision files

Using `hexdump` to view the contents of `md5-v1` and `md5-v2`. Are they different?

-> Yes

md5-v1	md5-v2
0000000 31d1 02dd e6c5 c4ee 3d69 069a af98 5cf9 0000010 ca2f 87b5 4612 ab7e 0440 3e58 fbb8 897f 0000020 ad55 0634 f409 02b3 e483 8388 7125 5a41 0000030 5108 e825 cdf7 9fc9 1dd9 f2bd 3780 5b3c 0000040 82d8 313e 3456 5b8f 6dae d4ac c936 c619 0000050 53dd b4e2 da87 fd03 3902 0663 48d2 a0cd 0000060 9fe9 4233 570f e87e 54ce 70b6 a880 1e0d 0000070 98c6 bc21 a8b6 9383 f996 2b65 f76f 702a 0000080	0000000 31d1 02dd e6c5 c4ee 3d69 069a af98 5cf9 0000010 ca2f 07b5 4612 ab7e 0440 3e58 fbb8 897f 0000020 ad55 0634 f409 02b3 e483 8388 f125 5a41 0000030 5108 e825 cdf7 9fc9 1dd9 72bd 3780 5b3c 0000040 82d8 313e 3456 5b8f 6dae d4ac c936 c619 0000050 53dd 34e2 da87 fd03 3902 0663 48d2 a0cd 0000060 9fe9 4233 570f e87e 54ce 70b6 2880 1e0d 0000070 98c6 bc21 a8b6 9383 f996 ab65 f76f 702a 0000080

3.2 Compute MD5 Collisions

Compute the md5 hash for both files. Are they different?

Show command line and result:

```
openssl dgst -md5 md5-v1
MD5(md5-v1)= 79054025255fb1a26e4bc422aef54eb4
openssl dgst -md5 md5-v2
MD5(md5-v2)= 79054025255fb1a26e4bc422aef54eb4
There are no difference between two hashes.
```

3.3 Investigation of MD5 Collision files

Preview the file `plain.jpg` and `ship.jpg`. Are there different?

Result:

```
Both of two picture are totally different.
```

3.4 Compute MD5 Collisions

Compute the md5 hash for `plain.jpg` and `ship.jpg`. Are they different?

Show command line and result:

```
[lab7@lab7 collisions_ex]$ openssl md5 < plane.jpg
(stdin)= 253dd04e87492e4fc3471de5e776bc3d
[lab7@lab7 collisions_ex]$ openssl md5 < ship.jpg
(stdin)= 253dd04e87492e4fc3471de5e776bc3d
```

3.5 Investigation of SHA0 Collision files

Dump the contents of `sha-v1` and `sha-v2`. Are they different?

sha-v1	sha-v2

0000000 66a7 02a6 5cb6 e7ff bc73 58f2 b326 b322 0000010 1bd0 971a 8426 53ef 3b3e 7f4b fe53 6237 0000020 c024 478e 59e9 bcb2 513b 8098 28b9 6865 0000030 7d24 0f11 f570 e2c5 59b4 a30c 5ff5 fe52 0000040 fdef 8f4c 8de6 35e8 9e32 3c60 1ec5 027f 0000050 5454 d110 1d67 8d10 a4f5 0d00 20cf 39a4 0000060 4949 2cd7 4fd1 03bb cf45 293a cd5d 9fa8 0000070 8f99 5587 9a2c b158 c3bd 8384 475e 8571 0000080 6ef9 be68 00bb d225 b6d2 df9e 7221 9841 0000090 88f6 1db4 9beb 1349 e6fb b596 7a45 99b3 00000a0 e121 59d7 891f 84de e857 3c61 9e6c 243b 00000b0 7928 d8d4 3b78 9c2d 93a9 a55e a726 c029 00000c0 df6e 01c5 e637 3093 97be 1260 5dcc 1cfe 00000d0 c414 8bc6 dbd1 cb3e 4324 598a 9ba0 b45d 00000e0 5635 0d3e df8b 2f57 b577 6530 f3ce 321f 00000f0 9ddc a0ba 4641 1e26 9499 5cbd 75d0 3d8e 0000100	0000000 66a7 02a6 5cb6 e7ff bc73 58f2 b326 b122 0000010 1bd0 d71a 8426 51ef 3bbe 7f4b fed3 6237 0000020 c0a4 458e 59e9 fcb2 513b 8098 2839 2865 0000030 7da4 0d11 f570 e0c5 5934 e30c 5f75 fc52 0000040 fd6f 8d4c 8d66 75e8 9e32 3e60 1e45 027f 0000050 54d4 d110 1de7 8d10 a4f5 0d00 20cf 39a4 0000060 4949 2cd7 4fd1 01bb cf45 693a cd5d 9da8 0000070 8f19 5587 9aac b158 c33d 8184 475e c571 0000080 6e79 fe68 00bb d025 b652 dd9e 72a1 d841 0000090 8876 1fb4 9b6b 1149 e67b f596 7ac5 99b3 00000a0 e1a1 19d7 899f 86de e857 3c61 9eec 263b 00000b0 79a8 98d4 3b78 9e2d 9329 a75e a7a6 8029 00000c0 df6e 03c5 e637 3093 973e 1060 5d4c 5cfe 00000d0 c414 89c6 db51 cb3e 43a4 598a 9b20 b45d 00000e0 5635 0d3e df8b 2f57 b577 6530 f3ce 301f 00000f0 9ddc e0ba 4641 1c26 9419 5cbd 7550 3d8e 0000100

3.6 Compute SHA0 Collisions

Compute the SHA hash for `sha-v1` and `sha-v2`. Are they different?

Show command line and result:
No

3.7 Investigation of SHA1 Collision files

Preview the file `shattered-1.pdf` and `shattered-2.pdf`. Are they different?

Result:
The header's color of each file is different

3.8 Compute SHA1 Collisions

Compute the SHA1 hash for `shattered-1.pdf` and `shattered-2.pdf`. Are they different?

Show command line and result:
Nope [lab7@lab7 collisions_ex]$ openssl sha1 < shattered-1.pdf (stdin)= 38762cf7f55934b34d179ae6a4c80cadccbb7f0a [lab7@lab7 collisions_ex]$ openssl sha1 < shattered-2.pdf (stdin)= 38762cf7f55934b34d179ae6a4c80cadccbb7f0a