消息摘要算法示例（python 和 go）

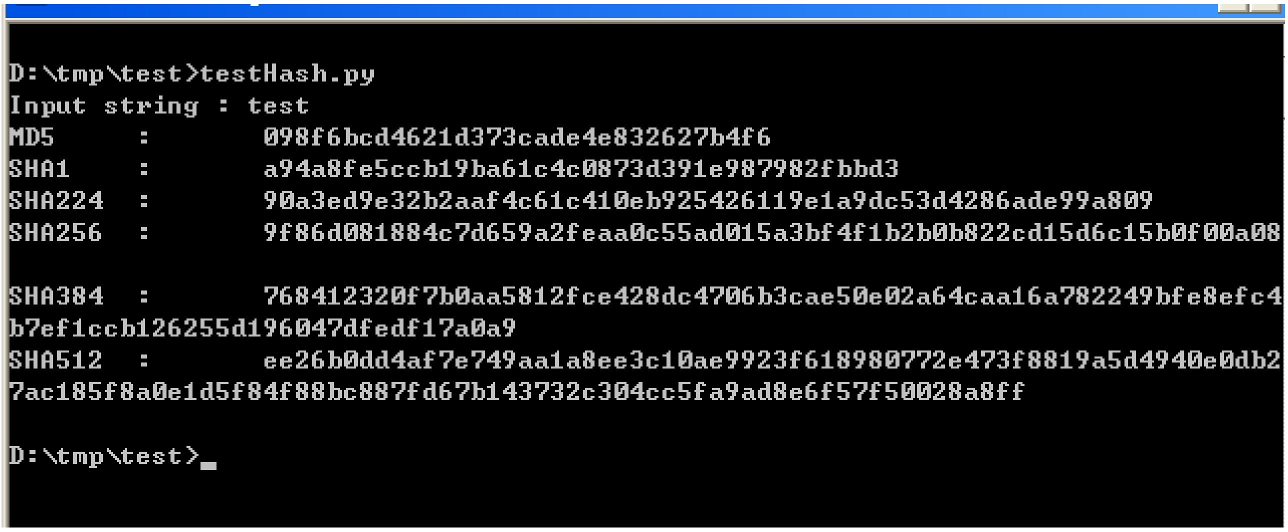
常用的消息摘要算法有 MD5 和 SHA，这些算法在 python 和 go 的库中都有，需要时候调用下就 OK 了，这里总结下 python 和 go 的实现。

一、python 消息摘要示例

代码如下：

```
#! /usr/bin/python
'''
     File       : testHash.py
     Author     : Mike
     E-Mail     : Mike_Zhang@live.com
'''

import hashlib

src = raw_input("Input string : ")

funcNameList = ["MD5","SHA1","SHA224","SHA256","SHA384","SHA512"]
funcMap = {
"MD5"           :       lambda cnt : hashlib.md5(cnt).hexdigest(),
"SHA1"          :       lambda cnt : hashlib.sha1(cnt).hexdigest(),
"SHA224"        :       lambda cnt : hashlib.sha224(cnt).hexdigest(),
"SHA256"        :       lambda cnt : hashlib.sha256(cnt).hexdigest(),
"SHA384"        :       lambda cnt : hashlib.sha384(cnt).hexdigest(),
"SHA512"        :       lambda cnt : hashlib.sha512(cnt).hexdigest()
}

for funcName in funcNameList :
       print funcName,"\t:\t",funcMap[funcName](src)
```

运行效果：



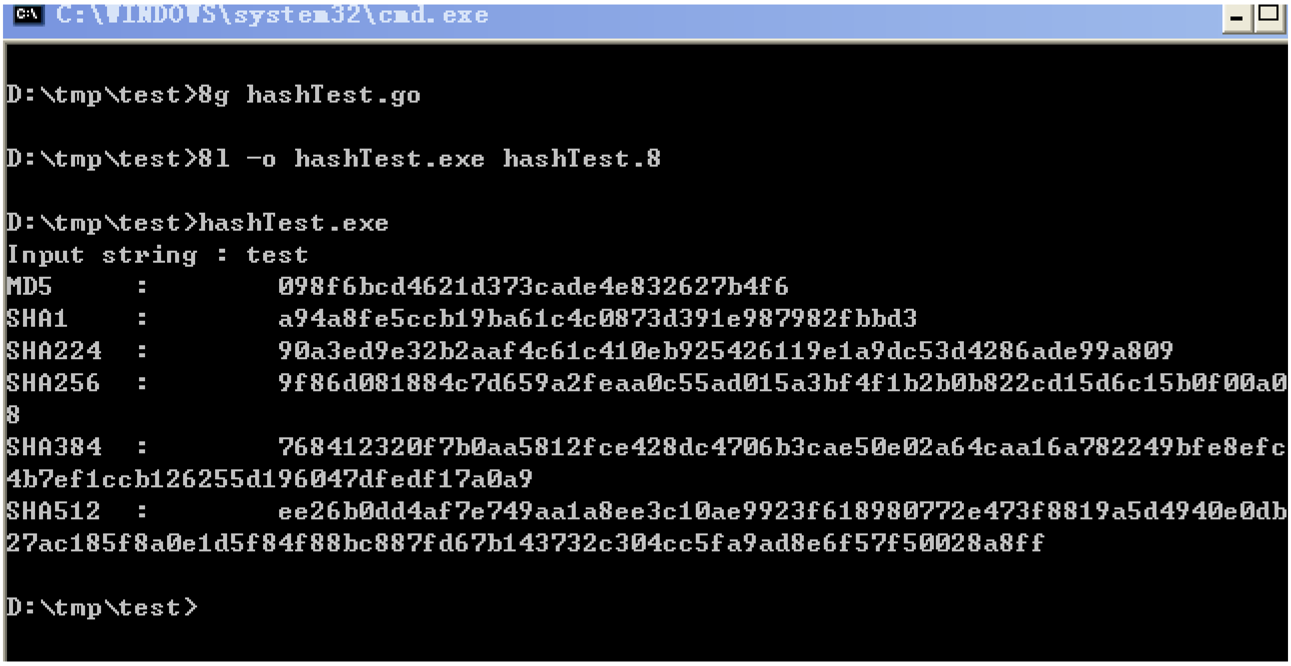
```
D:\tmp\test>testHash.py
Input string : test
MD5     :       098f6bcd4621d373cade4e832627b4f6
SHA1    :       a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
SHA224  :       90a3ed9e32b2aaf4c61c410eb925426119e1a9dc53d4286ade99a809
SHA256  :       9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08

SHA384  :       768412320f7b0aa5812fce428dc4706b3cae50e02a64caa16a782249bfe8efc4
b7ef1ccb126255d196047dfedf17a0a9
SHA512  :       ee26b0dd4af7e749aa1a8ee3c10ae9923f618980772e473f8819a5d4940e0db2
7ac185f8a0e1d5f84f88bc887fd67b143732c304cc5fa9ad8e6f57f50028a8ff

D:\tmp\test>_
```

二、go 语言消息摘要示例

代码如下：

```
/*
     File       : hashTest.go
     Author     : Mike
     E-Mail     : Mike_Zhang@live.com
*/

package main

import (
       "fmt"
       "crypto/md5"
       "crypto/sha1"
       "crypto/sha256"
       "crypto/sha512"
       "hash"
)

func main() {
       funcNameList :=
[]string{"MD5","SHA1","SHA224","SHA256","SHA384","SHA512"}
       funcMap := map[string]func(msg []byte) hash.Hash{
       "MD5"           :   func(msg []byte) hash.Hash{var h hash.Hash =
md5.New();h.Write(msg);return h},
       "SHA1"          :   func(msg []byte) hash.Hash{var h hash.Hash =
sha1.New();h.Write(msg);return h},
       "SHA224"        :   func(msg []byte) hash.Hash{var h hash.Hash =
sha256.New224();h.Write(msg);return h},
       "SHA256"        :   func(msg []byte) hash.Hash{var h hash.Hash =
sha256.New();h.Write(msg);return h},
       "SHA384"        :   func(msg []byte) hash.Hash{var h hash.Hash =
sha512.New384();h.Write(msg);return h},
       "SHA512"        :   func(msg []byte) hash.Hash{var h hash.Hash =
sha512.New();h.Write(msg);return h},
       }
       fmt.Printf("Input string : ")
       var msg1 string
       fmt.Scanf("%s",&msg1)
       for _,funcName := range funcNameList{
              fmt.Printf("%s \t:\t %x\n",funcName,funcMap[funcName]
([]byte(msg1)).Sum())
       }
}
```

运行效果：



```
D:\tmp\test>8g hashTest.go

D:\tmp\test>8l -o hashTest.exe hashTest.8

D:\tmp\test>hashTest.exe
Input string : test
MD5      :        098f6bcd4621d373cade4e832627b4f6
SHA1     :        a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
SHA224   :        90a3ed9e32b2aaf4c61c410eb925426119e1a9dc53d4286ade99a809
SHA256   :        9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a0
8
SHA384   :        768412320f7b0aa5812fce428dc4706b3cae50e02a64caa16a782249bfe8efc
4b7ef1ccb126255d196047dfedf17a0a9
SHA512   :        ee26b0dd4af7e749aa1a8ee3c10ae9923f618980772e473f8819a5d4940e0db
27ac185f8a0e1d5f84f88bc887fd67b143732c304cc5fa9ad8e6f57f50028a8ff

D:\tmp\test>
```

哈哈，是不是发现上面两组程序的代码结构相同啊，其实我就是想借助 python 来学习 go 语言的：先用 python 很轻巧的实现一个功能，我再考虑用 go 做一遍。这里总结下，方便以后使用。