实验报告

【实验目的】

1. 理解密码学$Hash$函数的安全性质、攻击方法、设计思想，掌握常用的密码学$Hash$函数的实现流程；
2. 理解消息认证码的设计思想，掌握基于$Hash$函数的消息认证码$Hash$的实现方法。

【实验环境】

1. 语言：C
2. 平台：clion 2021.2 版本

【实验内容】

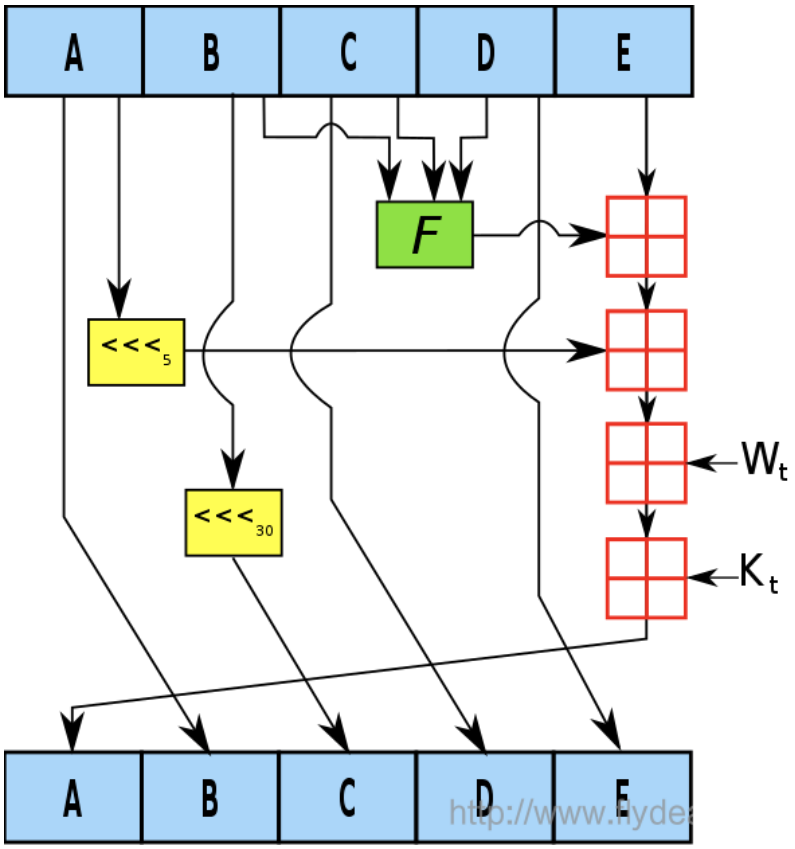
一、SHA-1

1. 算法流程

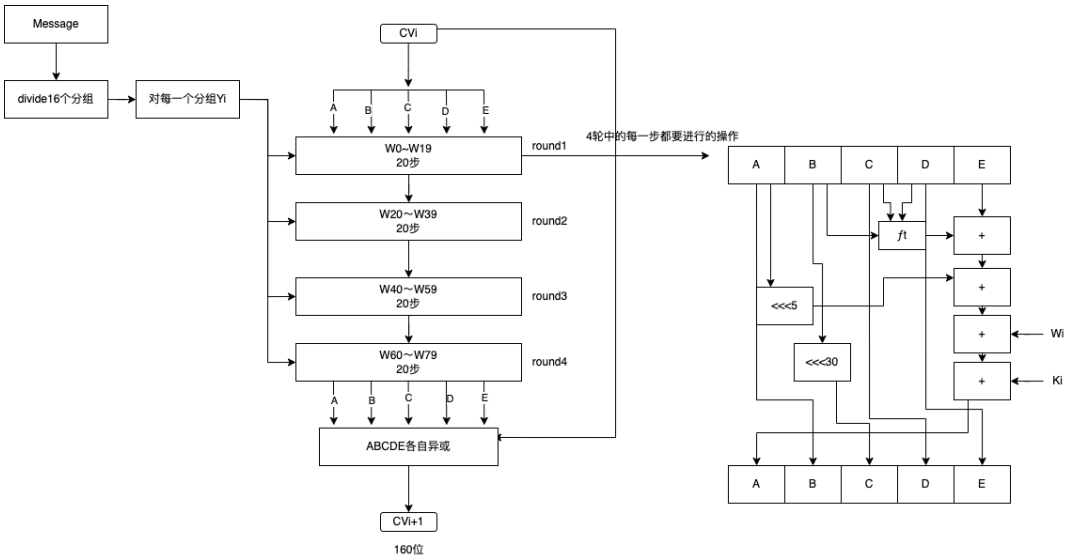
1. 对消息M进行填充
2. 开始循环，内容包括对消息的拓展以及压缩函数
3. 结束循环，输出最后一轮压缩函数的结果

- **流程图：**

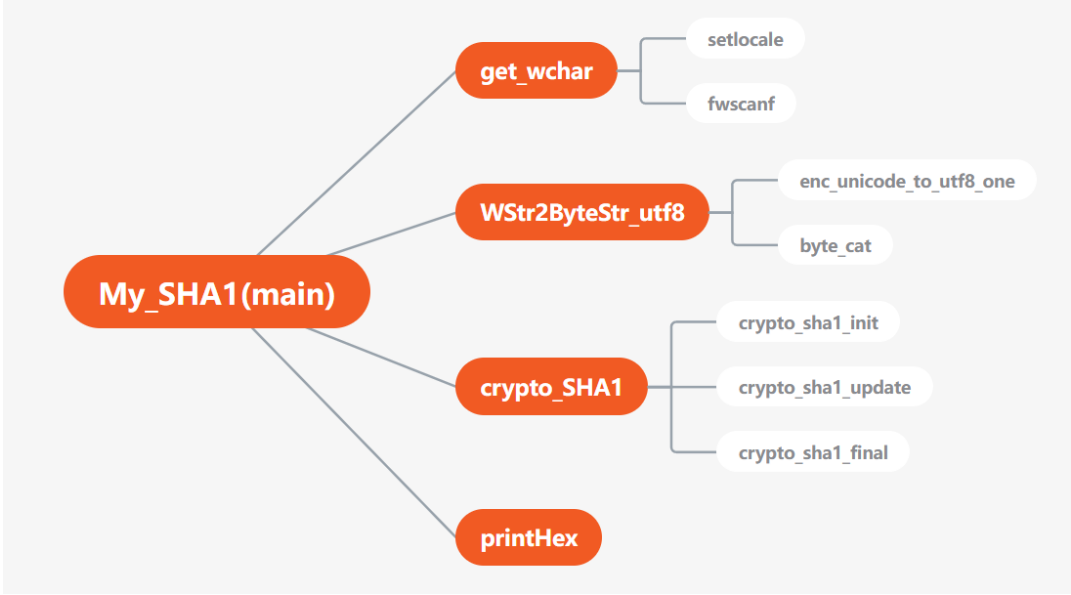
单轮流程图如下：



总流程图如下：



- **函数调用图：**



2. **测试样例及结果截图：**

本地测试样例的运行结果如下所示：

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--My_SHA1.exe"
this is the first SHA-1 test.
940e51a0efb1c3a684c6fa064abbc064de8975e1

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--My_SHA1.exe"
如果不出意料的话，每个题的样例都会出乎意料
f9374f3eaf64553abf2dc4c46356819a6675673a

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--My_SHA1.exe"
これは日本語の見本です
5ca635f776c6792000b378ac89f2cea6d7c1762e

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--My_SHA1.exe"
If you use hashlib.sha1 to finish these problems, you will get zero
68df2edd7bacf5aa74616321a9cee3fca823eb62

Process finished with exit code 0
```

3. **讨论与思考：**

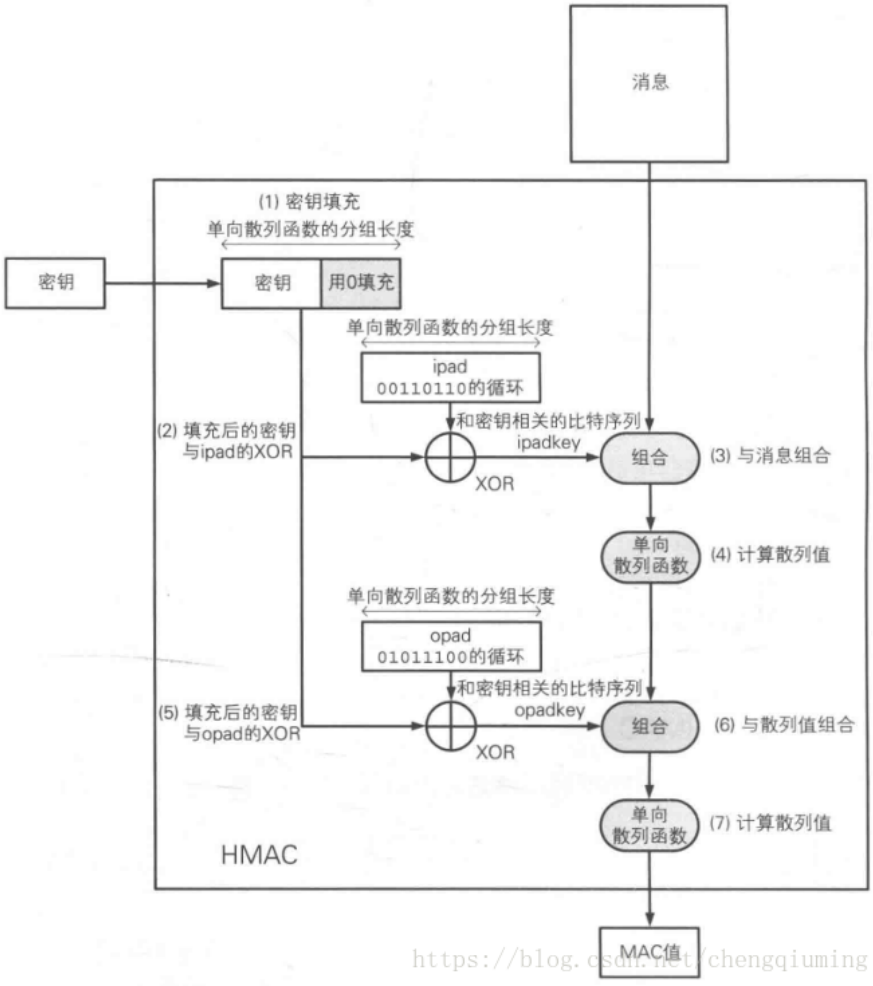
   消息的长度在代码中没有限制，因为如果过长，会只取后16位（十六进制）。而在RSA_OAEP使用的SHA-1是有相应限制的，长度为$2^{61}-1$。

二、HMAC-SHA1

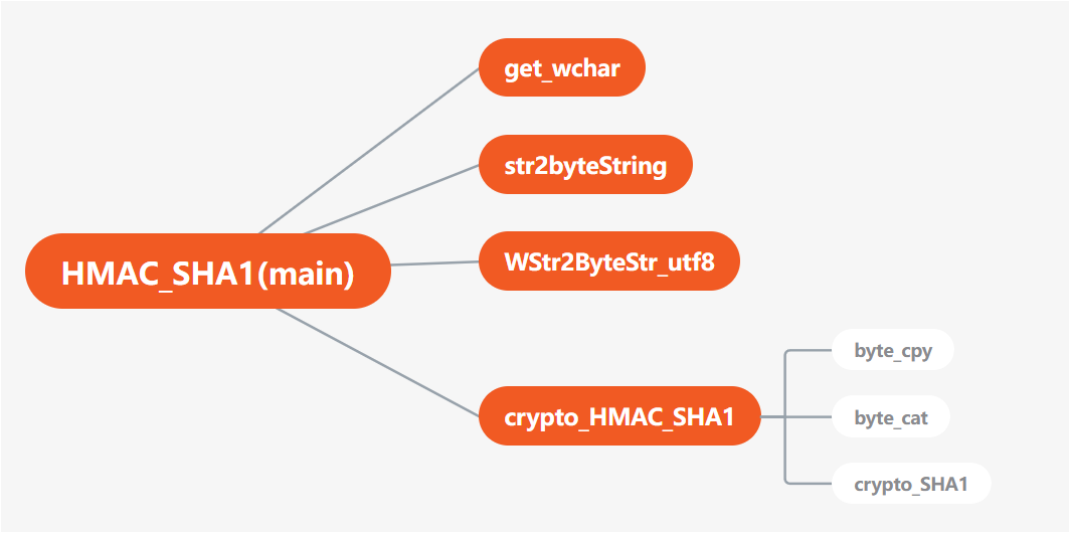
1. **算法流程**

   1. 对密钥进行填充
   2. 利用$ipad, opad$和密钥生成S_0, S_i
   3. 将S_i与M进行拼接，获取其哈希值tmp
   4. 将tmp与S_0进行拼接，获取其哈希值并进行输出

   - **算法流程图：**



   - **函数调用图如下：**



2. **测试样例及结果截图：**

   本地测试样例的运行结果如下所示：

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--HMAC_SHA1.exe"
000102030405060708090A0B0C0D0E0F10111213
this is the first HMAC test.
5429b4bc6b12a5dcdac8d0fe73560fc092ec473e

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--HMAC_SHA1.exe"
0123
Sample message for keylen<blocklen
cb5891545bb7e0267d87892cca82a3083c0a391e

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--HMAC_SHA1.exe"
1987
we're no strangers to love you know the rules and so do I a full commitment's what I'm thinking of y
 just wanna tell you how I'm feeling gotta make you understand never gonna give you up never gonna l
  you never gonna make you cry never gonna say good bye never gonna tell a lie and hurt you
f163cf4bb600ed1e09ae8cfdfa69a3a5b40e44b7

Process finished with exit code 0
```

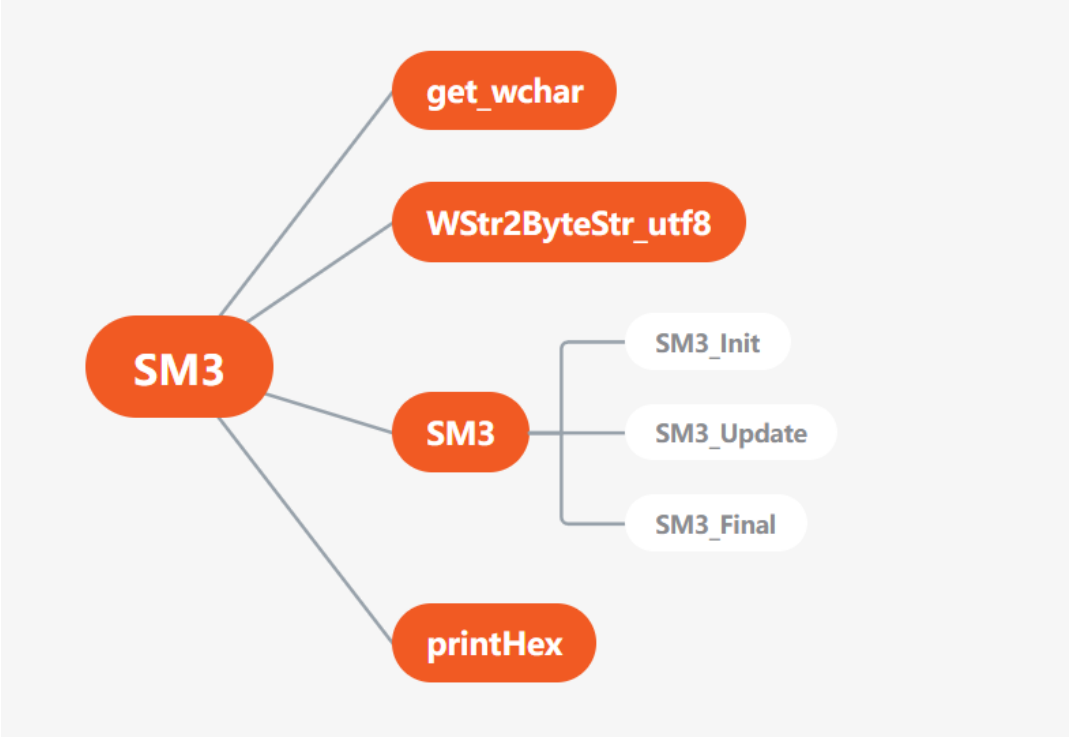
```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--HMAC_SHA1.exe"
000102030405060708090a0b0c0d0e0f101141aa
召唤增强G，特招蓝色喷流灵，喷流灵，启辉器，呼唤卡组红精灵，深海的歌姬，拥抱海皇重装兵，叠猫猫，去素材，六下四直接攻击，
 灶，番切灶，藏身在马桶里，淘气精灵，让不死之身觉醒，巨大喷流，他自动都送给你，哇哦耶~ 书呆神骑，天选号在此降临，卫星闪灵
11094da1838e85be5f3e56b59e99c7a2c61d22d7

Process finished with exit code 0
```

三、SM3 - 密码杂凑算法

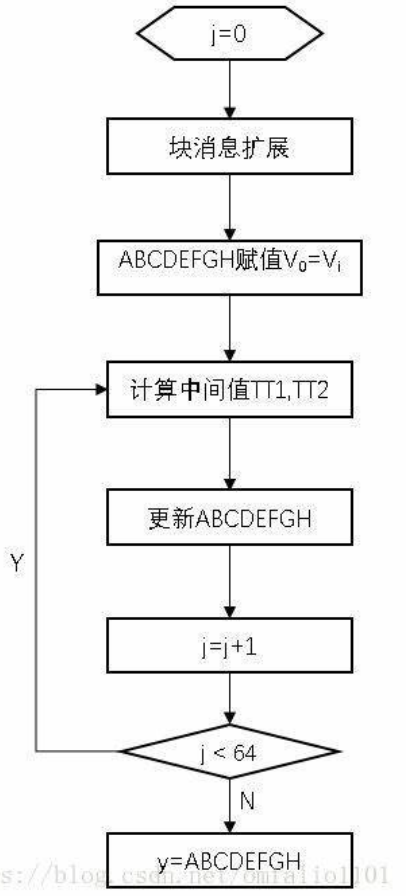
1. **算法流程**

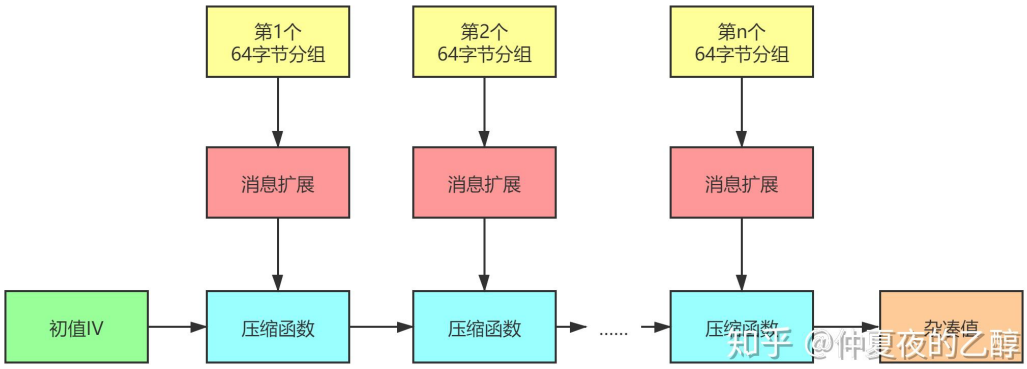
   1. 对消息进行编码
   2. 对消息进行预处理
   3. 开始循环，进行压缩
   4. 将最后一轮的压缩结果进行输出

- **函数调用图：**


get_wchar
WStr2ByteStr_utf8
SM3
SM3_Init
SM3
SM3_Update
SM3_Final
printHex

- **流程图：**


j=0
块消息扩展
ABCDEFGH赋值V0=Vi
计算中间值TT1,TT2
更新ABCDEFGH
Y
j=j+1
j < 64
N
y=ABCDEFGH



- **伪代码：**

算法 1 SM3

输入： m
输出： $y(list)$

function SM3(m)
  $M = padding(m)$
  $num = l//512$
  $v = [A, B, C, D, E, F]$
  for $i = 0 \rightarrow num$ **do**
    $v = CF(M[i * 64 : i * 64 + 64], v)$
  end for
end function

算法 2 CF

输入： m, v
输出： out

function CF(m)
  $w, w_1 = MessageExtension(m)$
  for $j = 0 \rightarrow 16$ **do**
    if $j < 16$ **then**
      $ss1 = ((a << 12) + e + (T[0] << (j \ mod \ 32))) \ mod \ (1 << 32) << 7$
    else
      $ss1 = ((a << 12) + e + (T[1] << (j \ mod \ 32))) \ mod \ (1 << 32) << 7$
    end if
    $ss2 = ss1 \oplus (a << 12)$
    $tt1 = (FF(a, b, c, j) + d + ss2 + w_1[j]) \ mod \ (1 << 32)$
    $tt2 = (GG(e, f, g, j) + h + ss1 + w[j]) \ mod \ (1 << 32)$
    $give \ value \ to \ A, B, C, D, E, F, G, H$
  end for
  $out = [a \oplus v[0], b \oplus v[1], c \oplus v[2], d \oplus v[3], e \oplus v[4], f \oplus v[5], g \oplus v[6], h \oplus v[7]]$
  return out
end function

2. 测试样例及结果截图：

本地测试样例的运行结果如下所示：

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--SM3.exe"
Можете дори да видите примери за български
68fcac3c01480a28c81b156497ce9ba45830e80d34e567a720421214395f24dc

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--SM3.exe"
这次题很简单，数据也没有任何限制，那为什么不放飞自我呢
2912eefd2d16d65541fd88fb331ed51aea05775e3e7f14d619b016a124491f5a

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--SM3.exe"
listen me say thank thank you, because have you, warmed the four seasons, thank thank you, than
67e296197612479b386335ba4cff5e2495643411941e5845ee8ed72c06f53f5f

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\cmake-build-debug\Exp9--SM3.exe"
this is the first SM3 testcase.
1c7d1fcf91f37a2ecb8877b5896d3474010784a75cdb1d392375029c4469e653

Process finished with exit code 0
```

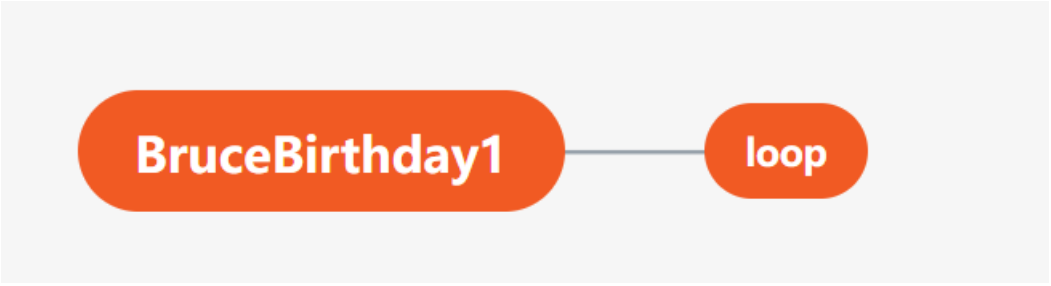
四、【选做】SHA-1第一类生日攻击

1. **算法流程**

   - **原理：**

     持续的遍历英文字母的组合，直到生成满足要求的结果。

   - **函数调用图：**



2. **测试样例及结果截图：**

本地测试样例的运行结果如下所示：

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\venv\Scripts\python.exe" "E:/E_drive/clion/Project
 List/cryptography/Crypto_Experiment/Exp9/资料/py_code/BruceBirthday1.py"
4
d2ada4b1046caec75e7ae2dcd48c89f0eaeb6d794
d2ada4b1046caec75e7ae2dcd48c89f0eaeb6d79

CAcct

Process finished with exit code 0
```

```
"E:\E_drive\clion\Project List\cryptography\Crypto_Experiment\venv\Scripts\python.exe" "E:/E_drive/clion/Project
 List/cryptography/Crypto_Experiment/Exp9/资料/py_code/BruceBirthday1.py"
8
b9103d6a789ce45dcdc9dfd973d3c7a789d6af468
b9103d6a789ce45dcdc9dfd973d3c7a789d6af46

CAci4

Process finished with exit code 0
```

查看历史提交

评测编号	提交时间	提交状态	代码语言	最大运行时间	最大运行内存	详细信息
25502	2022-06-01 16:34:31	Accepted	Python	3015ms	12544KB	

3. **讨论与思考：**

需要手动的试字符串的长度，来保证时间以及$n = 24$时的结果可以求出。

五、【选做】SHA-1第二类生日攻击

1. **算法流程**

   穷举法yyds！！！！！

   - **函数调用图：**

2. **测试样例及结果截图：**

   本地测试样例的运行结果如下所示：

七：讨论与思考

对SHA-1，HMAC以及SM3的流程有了更好的理解。