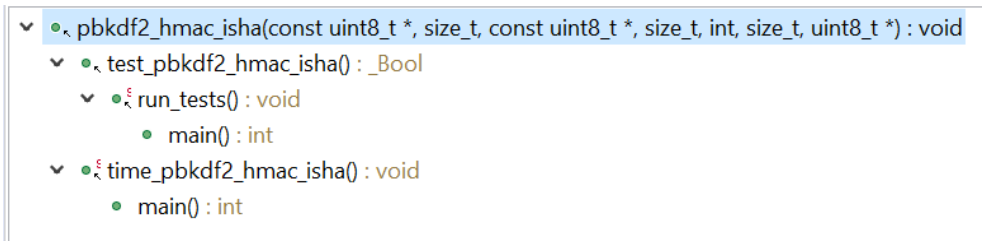
@File: Technical Memo for the Optimization Assignment (ISHA)
@Author: Harshwardhan Singh (harshwardhan.singh@colorado.edu)

CALL STACK

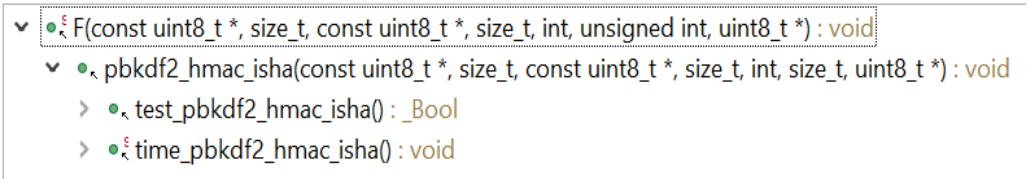
pbkdf2_hmac_isha() Function –

'I' is calculated to find the number of 20 is to 8 bits blocks. Function 'F' is called for each block of DK where pass, pass_len, salt, salt_len, iteration count and block index are passed to compute the block.
It combines the blocks and gives the first dkLen octets to produce a derived key DK. The resultant data is stored in DK (Derived Key). The derived key of 'dkLen' bytes is stored in DK.



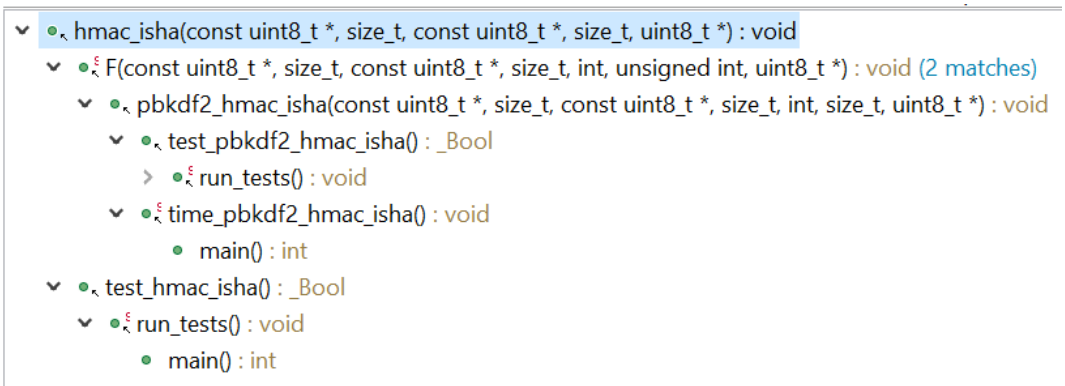
F() Function –

The data of 'salt' is copied into 'saltplus' and block index is appended in 4 bytes big endian by using suitable operations as ARM is little endian. Function hmack_isha() has been called inside this F() function with pass, pass_len, saltplus, salt_len passed as arguments. This function array hashes the 'salt' and 'pass' to make it into block length array. This iteration is added at the end of the array and hashing is done 4096 times.



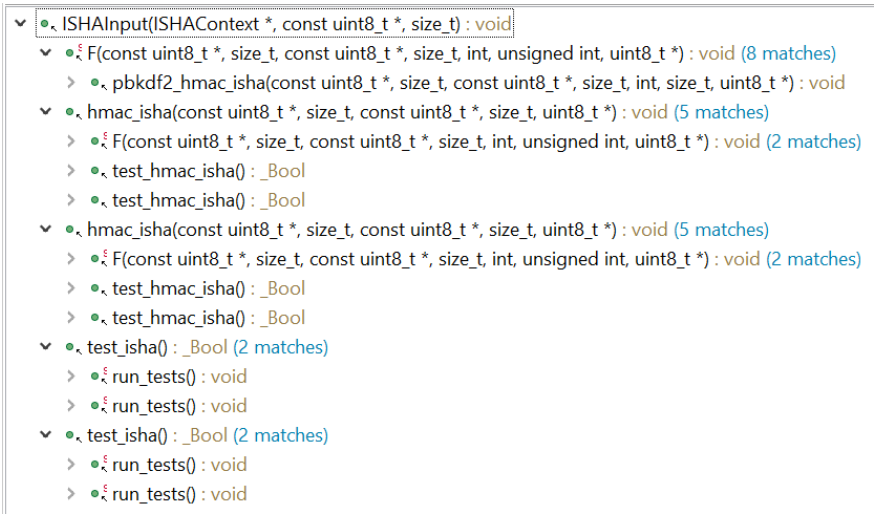
hmac_isha() Function –

Arrays or strings ipad and opad are initialized as inner pad and outer pad.
If (key_len>ISHA_BLOCKLEN), key is reset, putting it under the 64-byte length. But, if (key_len<=ISHA_BLOCKLEN), data of key is copied in keypad that is of size ISH_BLOCKLEN bytes and it is zero padded.
ipad is the byte 0x36 repeated 64 times and opad is the byte 0x5C repeated 64 times. Zero is appended at the end of key to create a B-byte string.
XOR operation is performed on 64 byte string with ipad and the operation is performed on 64 byte string with opad. Performs inner isha, over ipad and message (saltplus), and the array. Again, performs outer isha, over opad and inner_digest. The resultant 20-byte key is returned in digest.



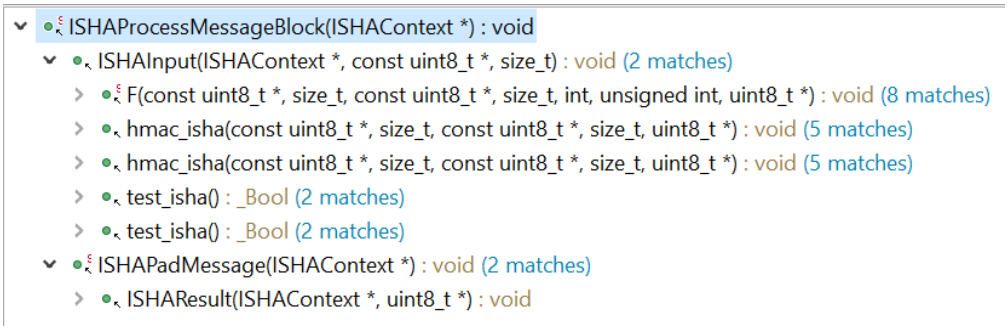
ISHAInput() Function –

This function is used to push bytes into the ISHA hashing algorithm, the ISHA algorithm can has upto 2^{61} bytes.
Takes an array of bytes for performing ISHA hash and keeps monitoring the current message length in bits using Length_Low and Length_High.
Whenever the message is added to MBlock (64 bytes), Length_Low is incremented by 8 to store the length in bits.
If message is too long for hashing, the following data is put on the next first 64 bytes and the corrupted flag is set.
To process the data, the ISHAProcessMessageBlock() is called.



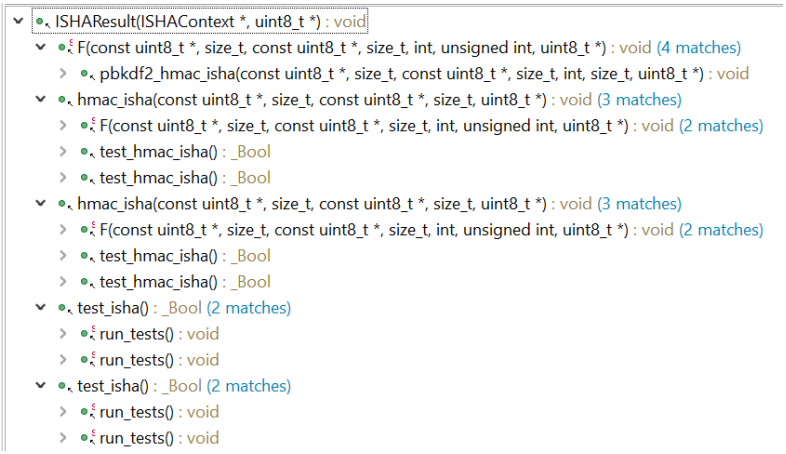
ISHAProcessMessageBlock() Function –

In this function, hashing and manipulation is done with 512 bites i.e. 64 bytes in the message digest which is an output to this function that is 160 bits.



ISHAResult() Function –

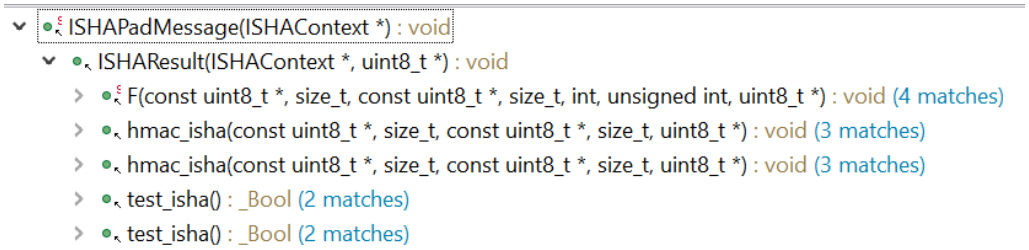
This function is called once all the bytes have been put into the algorithm. This function gives the ISHA hash as an output that is 160 bits (20 bytes).
Data of message digest (MD[i]) are stored in digest out of 20 bytes using suitable operators and shifting.
ISHAPadMessage() is called within this function and computed flag is set to '1'.



ISHAPadMessage() Function –

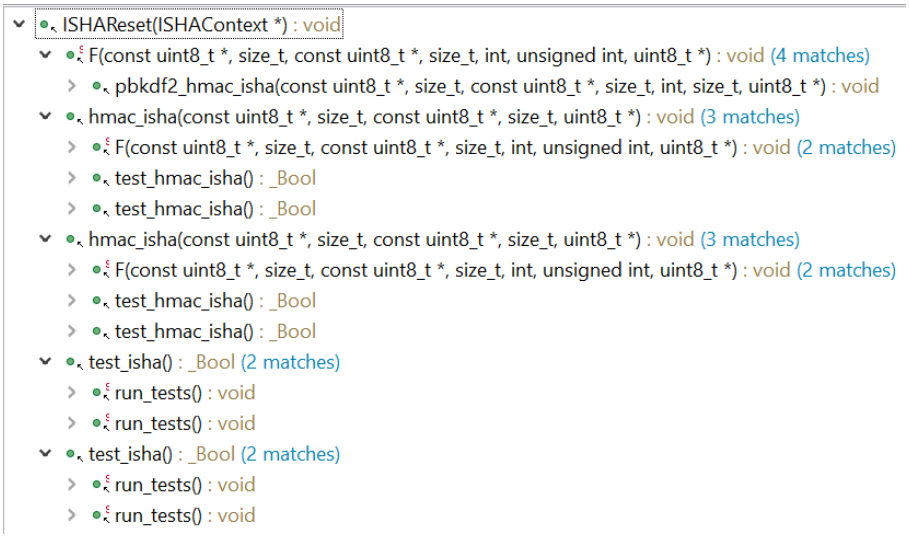
The message is padded by following the mentioned rule i.e. the message must be padded to an even 512 bits (64-bytes). The first padding bit must be '1'. The last 64 bits represent the length of the message. All other bits in between must be set as '0'.

ISHAProcessMessageBlock() function is called accordingly. If the current message block is too small to hold the initial padding bits and length, the current block is padded, processed and then continues padding into a second block.



ISHAReset() Function –

Whenever ISHA_Reset() is called, it acts as an initialization function as it reset all the parameters namely: message_index to '0', Length_Low and Length_High to '0', MD[1 to 5] to some constant hex values, computed and corrupted flags are also set to '0'.



Timing Profile (before optimization)

Single call to each of the following functions –

#Note: all the timings are approximate

Functions	Time (msec)
pbkdf2_hmac_isha()	8744
F()	2914
hmac_isha()	0.711
ISHAReset()	0.024
ISHAResult()	0.013
ISHAInput()	0.1626
ISHAProcessMessageBlock()	0.058
ISHAPadMessage()	0.086

Number of calls of each function

Functions	Number of calls
pbkdf2_hmac_isha()	1
F()	3
hmac_isha()	12288
ISHAReset()	24576
ISHAResult()	24576
ISHAInput()	49152
ISHAProcessMessageBlock()	49152
ISHAPadMessage()	24576

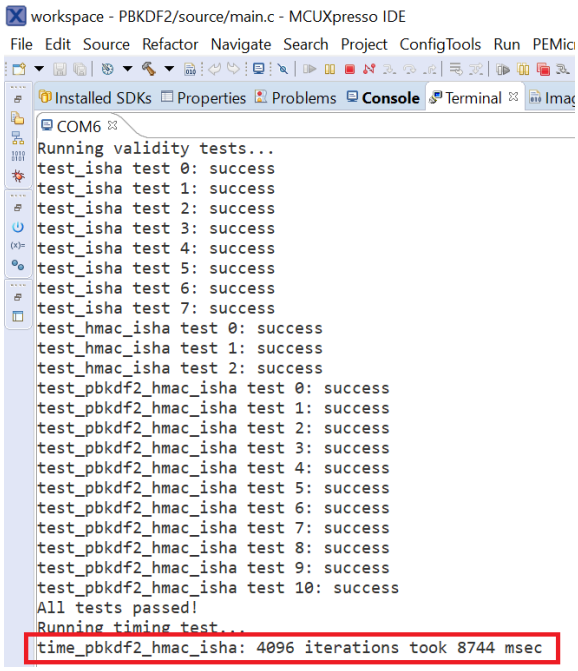
Size of the .text segment (before optimization) – 21,056 bytes (0x00005240)

```
Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00005240  00000000  00000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000008  1ffff080  00005240  0001f080  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          0000017c  1ffff088  1ffff088  0002f000  2**2
                  ALLOC
```

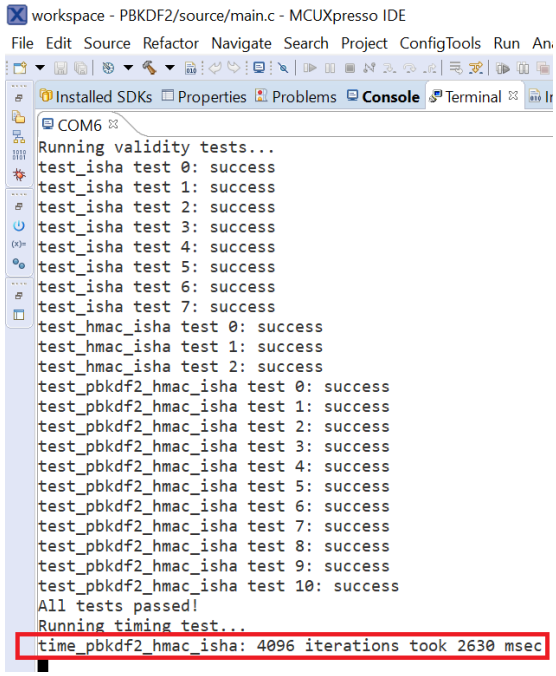
Size of the .text segment (after optimization) – 20,260 bytes (0x00004F24)

```
Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00004f24  00000000  00000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000008  1ffff080  00004f24  0001f080  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          0000017c  1ffff088  1ffff088  0002f000  2**2
                  ALLOC
  3 .mtb_buffer_default 00000080  1ffff000  1ffff000  0002f000  2**7
                  ALLOC
  4 .uninit_RESERVED 00000000  1ffff080  1ffff080  0001f088  2**2
                  CONTENTS
```

Total time taken (before optimization) - 8744 msec



Total time taken (after optimization) - 2630 msec



Function Flow–