

Assignemnt 2

Yunze Li

2016-1-31

1. Introduction and Functions

In this assignment, I used the *gcrypt* library in Linux operating system to implement many functions including: encryption, decryption, hashing and digital signature. This program timed each function or algorithm by perform it 100 times and calculate the median and mean of the 100 results. During each iteration, the key will be generated randomly but the generation time will not included in the calculated time.

The encryption algorithm I used including: AES128 CTR Mode, AES256 CTR Mode, RSA1024, RSA4096, HMAC MD5, HMAC SHA1 and HMAC SHA256. At last, this program also generate a digital signature using HMAC SHA256 and RSA4096.

For all these four ciphers and three hash algorithms, I snapshotted each output I got when I run this program in Linux environment, I put all graphs in the last section, please go to section 5 to check all results.

2. Development Environment and Machine Version

Development Environment: Linux Ubuntu 14.04 LTS

Memory: 3.9 GiB

Processor: Intel® Core™ i7-3720QM CPU @ 2.60GHz x2

Graphics: Parallels using NVIDIA GeForce GT 650M OpenGL Engine

OS-Type: 64-bits

3. TestFile(input_file) Information

Since the Memory in my virtual machine is limited, I only can allocate a limited space for the whole program to run. So for now it can only read some small file(like the input_file in this tgz file) and runs perfectly. To meet the assignment 2's requirement, first I modified my code in the reading file part. When a file is read, the program will access to the address that maps this file and then read the necessary blocks instead of copy the whole file's data into the secure memory. For the 100MB file, my solution is using a dividing tool, cutting it into pieces and then running my program to encrypt and decrypt each piece. In this way, the 100MB file can be solved successfully. I will keep working on modifying my program and find more effective methods to deal with the 100MB file problem.

4. Rank Ordering

After testing each function by this program, the rank ordering shows:

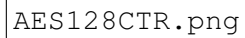
AES128<AES256<<RSA1024<<RSA4096

HMAC SHA1<<HMAC MD5<HMAC SHA256

Which is same as what we learned at class.

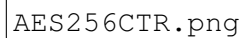
5. Result Snapshot

Here are the snapshots of each function:

A rectangular box containing the text "AES128CTR.png".

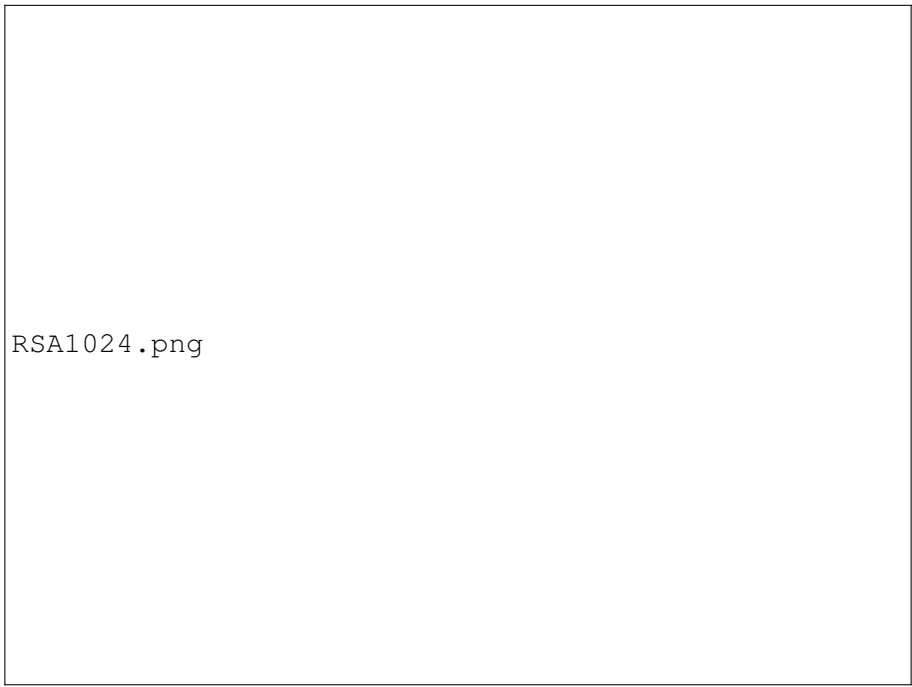
AES128CTR.png

AES128 CTR Mode

A rectangular box containing the text "AES256CTR.png".

AES256CTR.png

AES256 CTR Mode



RSA1024.png

A large rectangular box representing the RSA1024 Mode diagram. The box is empty except for the text 'RSA1024.png' in the top-left corner.

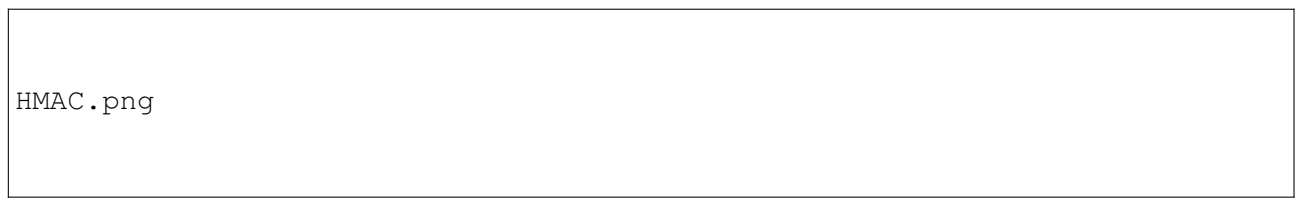
RSA1024 Mode



RSA4096.png

A large rectangular box representing the RSA4096 Mode diagram. The box is empty except for the text 'RSA4096.png' in the top-left corner.

RSA4096 Mode



HMAC.png

A wide rectangular box representing the HMAC Mode diagram. The box is empty except for the text 'HMAC.png' in the top-left corner.

HMAC(Three Hashing Algorithm in together)

DigitalSignature.png

Digital Signature