Understanding the Galois/Counter Mode (GCM)

HW3-CNS Sapienza

Manuel Ivagnes 1698903

21/11/19

1 Introduction

GCM is an operating mode used for encryption and decryption with symmetric-key cryptographic block ciphers, which introduce an authenticated encryption, so that, it provides both data authenticity and confidentiality. More informations about the operating mode are contained in the slides.

To understand the behaviour of this new operating mode in real scenario, it has been compared with the CBC mode, using for both the AES cipher. CBC has less operations to do, but GCM can use the parallel processing also during the encryption, and the pipeline always. It is so interesting to see how and if, increase the security, increase also the total time required by the operations.

2 Measures

As for the other homework I started by trying the operations by shell, but after a failed test I found on the OpenSSL documentation this statement: *The enc program does not support authenticated encryption modes like CCM and GCM, and will not support such modes in the future.*

Given this fact I decided to switch to the C language (the bash code is still provided in the file *code1-.sh* to use with libreSSL).

The OpenSSL library for C provides two interfaces, the Standard Interface and the EVP interface. For these test I have used the high level interface EVP, which provides a better support for the AES chiper. Moreover, the

standard library does not support the authenticated encryption.

Since the OpenSSL wiki provides useful standard implementations for decryption and decryption, I decided to use them for the homework. All the C code is contained in *code2-.c*.

This time for the tests I have used first two text files, txt format, and then two pdf files. Each test has been repeated 3 times. To notice that, the encryption and decryption output are just saved in the local memory of the execution, there are not output files, this obviously decreases the amount of time needed by the operations.

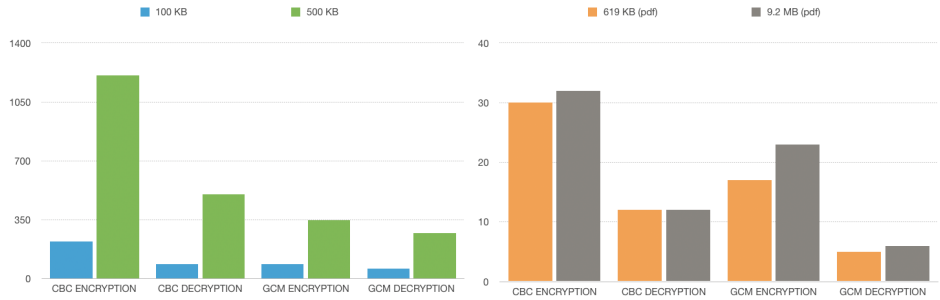
1. Results test txt size 100 KB
    - Operating Mode 1: CBC
        - Encryption time average: 0.000221s
        - Decryption time average: 0.000088s
        - Speed-ratio: 2.5113
    - Operating Mode 2: GCM
        - Encryption time average: 0.000088s
        - Decryption time average: 0.000061s
        - Speed-ratio: 1.4426
2. Results test txt size 500 KB
    - Operating Mode 1: CBC
        - Encryption time average: 0.001207s
        - Decryption time average: 0.000500s
        - Speed-ratio: 2.4140
    - Operating Mode 2: GCM
        - Encryption time average: 0.000349s
        - Decryption time average: 0.000272s
        - Speed-ratio: 1.2830
3. Results test pdf size 619 KB
    - Operating Mode 1: CBC
        - Encryption time average: 0.000030s
        - Decryption time average: 0.000012s

    - Speed-ratio: 2.5000
- Operating Mode 2: GCM
    - Encryption time average: 0.000017s
    - Decryption time average: 0.000005s
    - Speed-ratio: 3.4000

4. Results test pdf size 9.2 MB

- Operating Mode 1: CBC
    - Encryption time average: 0.000032s
    - Decryption time average: 0.000012s
    - Speed-ratio: 2.6666
- Operating Mode 2: GCM
    - Encryption time average: 0.000023s
    - Decryption time average: 0.000006s
    - Speed-ratio: 3.8333

DIMENSIONE	100 KB	500 KB	619 KB (PDF)	9.2 MB (PDF)
CBC ENCRYPTION	221	1207	30	32
CBC DECRYPTION	88	500	12	12
GCM ENCRYPTION	88	349	17	23
GCM DECRYPTION	61	272	5	6



3 Conclusions

It it important to notice that, the short execution time for the pdf file is given by the method by which the c-code read them, it is not properly correct but it is fine for this comparison.

The CBC operating mode confirm the average 2.5 speed-ratio seen last time, given by the slow encryption.

The GCM operating mode gave particular results, mostly considering the pdf files, where the time for the operations is short. In this case the high speed ratio is probably given by the preprocessing of the decrypt operations. When the time needed is longer, instead, parallelizing also the encryption, the speed-ratio is closer to 1.

In general GCM is faster than CBC (at least considering only encryption and decryption without a real authentication).

References

[1] Course slides

[2] https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

[3] https://en.wikipedia.org/wiki/Galois/Counter_Mode

[4] https://wiki.openssl.org/index.php/EVP_Authenticated_Encryption_and_Decryption

[5] https://wiki.openssl.org/index.php/EVP_Symmetric_Encryption_and_Decryption