LAB ASSIGNMENT: ENTROPY

CRYPTOGRAPHY AND COMPUTER SECURITY

Ana I. González-Tablas Ferreres
José María de Fuentes García-Romero de Tejada
Lorena González Manzano
Sergio Pastrana Portillo
UC3M | COMPUTER SECURITY LAB (COSEC) GROUP

TOOLS

ENT. Available in http://www.fourmilab.ch/random/

- Windows: Decompress it within a folder.
- Unix: Decompress, compile it using *make* and execute it using *./ent.*
- Execution:
  - Move to the directory where the executable file is located. Then: ent "name of the file to analyse"

OPENSSL. Available for Windows and Linux. For Windows: https://wiki.openssl.org/index.php/Binaries

- In Windows the path to this program should be included in the environment variable PATH. Use the command: *set PATH=%PATH%;"PATH DONDE INSTALE OPENSSL"/bin*

INTRODUCTION

In cryptography, one of the requirements of a cryptographic algorithm corresponds to the achievement of a random output. By contrast, if the output is not random, cryptanalysis can be easier and third parties can take advantage of this issue. Unfortunately, there is no a concrete definition of randomness and it is impossible to be completely certain about the randomness of a set of data. To mitigate this problem, along the time, some tests have been developed to empirically measure randomness. Although these tests cannot absolutely certify the existence of randomness, they can identify sets of data that are not, though they look like random. In this assignment we are going to analyze the randomness in respect to different files (encrypted and decrypted) and sets of pseudo-random data. Moreover, we are going to identify consequences of randomness regarding operations such as compression. Finally, we realize that some files, despite being files with high entropy, they are far from been random

(ex.: jpg. files).

Please, read carefully the documentation in http://www.fourmilab.ch/random/

Example of the output:

> ENT performs a variety of tests on the **stream of bytes** in *infile* (or standard input if no *infile* is specified) and produces output on the standard output stream. Example:
>
> **Entropy** = 7.980627 bits per character. (max. 8)
>
> **Optimum compression** would reduce the size of this 51768 character file by 0 percent.
>
> **Chi square** distribution for 51768 samples is 1542.26, and randomly would exceed this value less than 0.01 percent of the times.
>
> **Arithmetic mean value of data bytes** is 125.93 (127.5 = random).
>
> **Monte Carlo value for Pi** is 3.169834647 (error 0.90 percent).
>
> **Serial correlation coefficient** is 0.004249 (totally uncorrelated = 0.0).

EXERCISES

Exercise 1:

a) Download the following files (if you cannot download any of them, substitute it for other of the same type):

⇒ **Type doc:**

https://d9db56472fd41226d193-1e5e0d4b7948acaf6080b0dce0b35ed5.ssl.cf1.rackcdn.com/spectools/docs/wd-spectools-word-sample-04.doc

⇒ **Type c:**

hhttps://www.sanfoundry.com/c-program-replace-line-text-file/

(copiar el primer programa en un fichero y poner extensión .c)

⇒ **Type jpeg:** http://www.stallman.org/IMG_5884.JPG

⇒ **Type gif:** http://www.ritsumei.ac.jp/~akitaoka/cogwhee1.gif

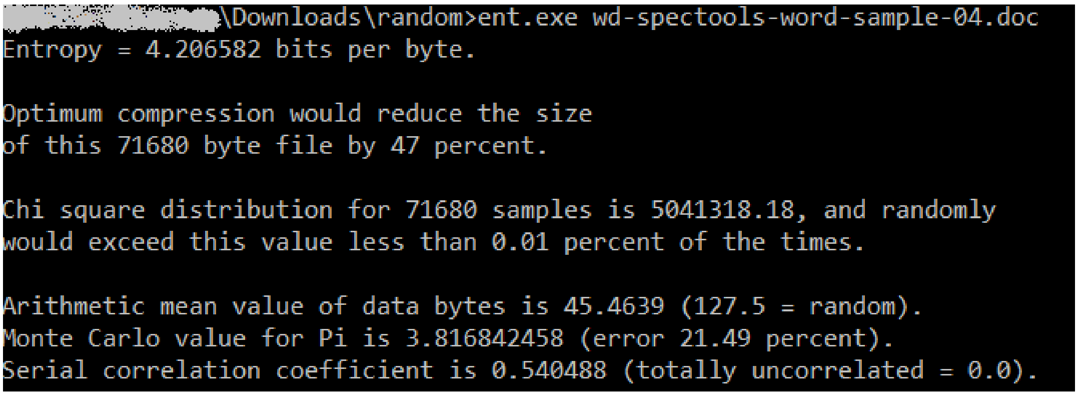
⇒ **Type bmp:** http://www.websiteoptimization.com/secrets/web-page/6-4-balloon.bmp

b) Execute ENT using the previous files as input and analyze the results.

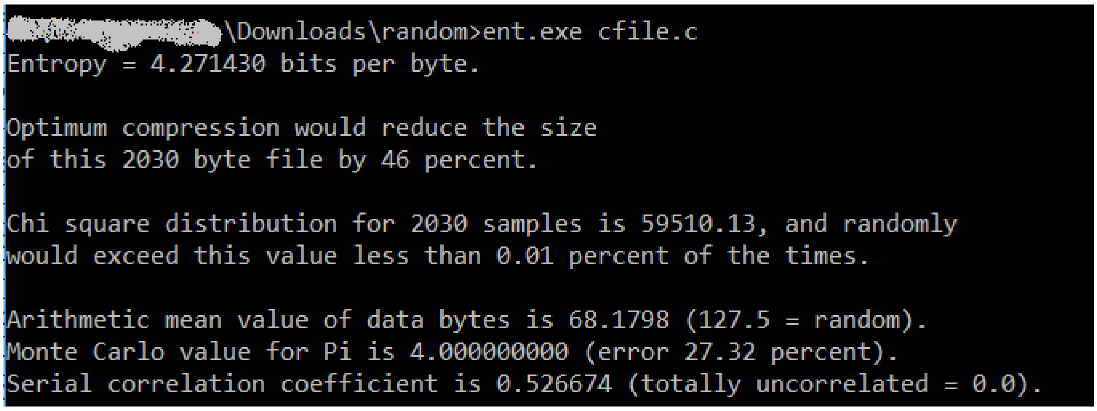
c) According to the analyses carried out in a), answer to the following question: Are results what you expected?

Solution:

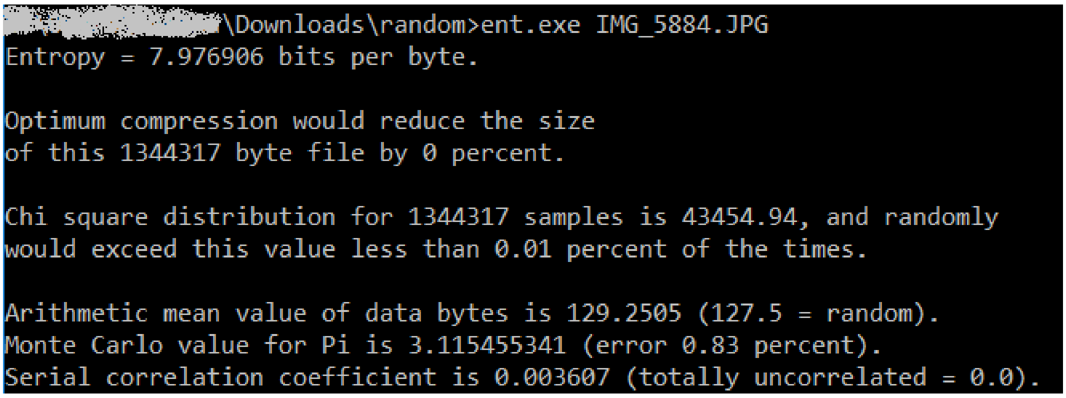
a) The result of applying ENT over the files is the following:



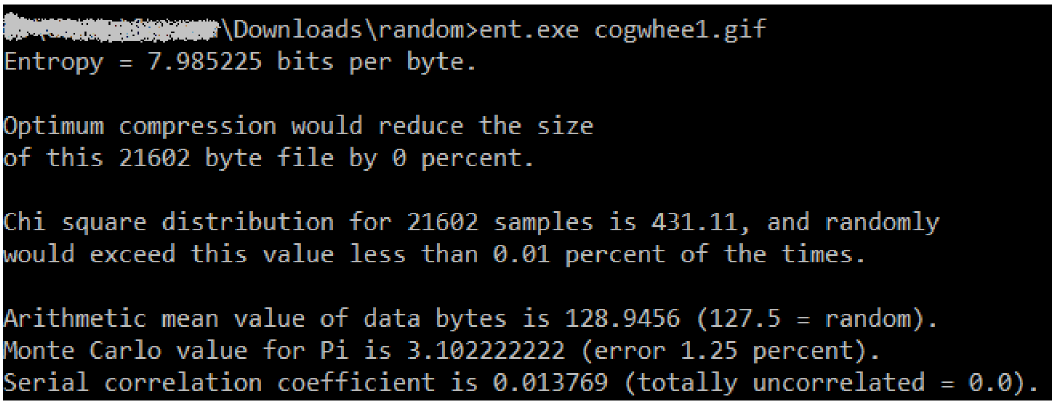
```
[redacted]\Downloads\random>ent.exe wd-spectools-word-sample-04.doc
Entropy = 4.206582 bits per byte.

Optimum compression would reduce the size
of this 71680 byte file by 47 percent.

Chi square distribution for 71680 samples is 5041318.18, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 45.4639 (127.5 = random).
Monte Carlo value for Pi is 3.816842458 (error 21.49 percent).
Serial correlation coefficient is 0.540488 (totally uncorrelated = 0.0).
```



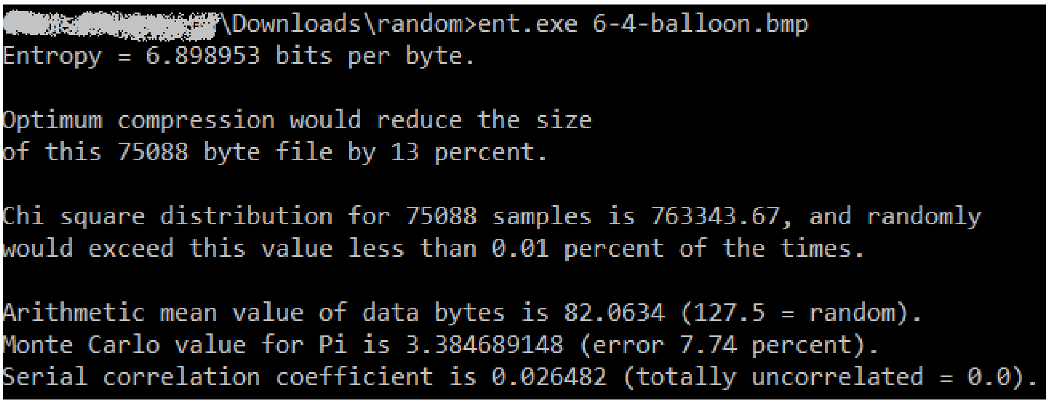
```
[redacted]\Downloads\random>ent.exe cfile.c
Entropy = 4.271430 bits per byte.

Optimum compression would reduce the size
of this 2030 byte file by 46 percent.

Chi square distribution for 2030 samples is 59510.13, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 68.1798 (127.5 = random).
Monte Carlo value for Pi is 4.000000000 (error 27.32 percent).
Serial correlation coefficient is 0.526674 (totally uncorrelated = 0.0).
```



```
[redacted]\Downloads\random>ent.exe IMG_5884.JPG
Entropy = 7.976906 bits per byte.

Optimum compression would reduce the size
of this 1344317 byte file by 0 percent.

Chi square distribution for 1344317 samples is 43454.94, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 129.2505 (127.5 = random).
Monte Carlo value for Pi is 3.115455341 (error 0.83 percent).
Serial correlation coefficient is 0.003607 (totally uncorrelated = 0.0).
```



```
[redacted]\Downloads\random>ent.exe cogwhee1.gif
Entropy = 7.985225 bits per byte.

Optimum compression would reduce the size
of this 21602 byte file by 0 percent.

Chi square distribution for 21602 samples is 431.11, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 128.9456 (127.5 = random).
Monte Carlo value for Pi is 3.102222222 (error 1.25 percent).
Serial correlation coefficient is 0.013769 (totally uncorrelated = 0.0).
```



```
[redacted]\Downloads\random>ent.exe 6-4-balloon.bmp
Entropy = 6.898953 bits per byte.

Optimum compression would reduce the size
of this 75088 byte file by 13 percent.

Chi square distribution for 75088 samples is 763343.67, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 82.0634 (127.5 = random).
Monte Carlo value for Pi is 3.384689148 (error 7.74 percent).
Serial correlation coefficient is 0.026482 (totally uncorrelated = 0.0).
```

According to results, there is not a single random file because not all tests are passed. However, codification algorithms like .jpg, .gif and .bmp present better results because they pass a greater number of tests.

b) Results are fair, text files are expected to pass a lower number of tests because they are more redundant (among other issues). However, picture files should commonly have higher entropy.

Exercise 2:

a) Use OpenSSL manual (https://www.openssl.org/docs/man1.0.2/) and explain how the following commands work:

⇒ *openssl rand -out r1000 -rand FILE -base64 1000*

⇒ *openssl rand -out r1000000 -rand FILE -base64 1000000*

FILE can be linked to any type of file, for instance “CA.pl”

b) Execute the previous commands and analyze the file using ENT. What can you conclude?

Solution:

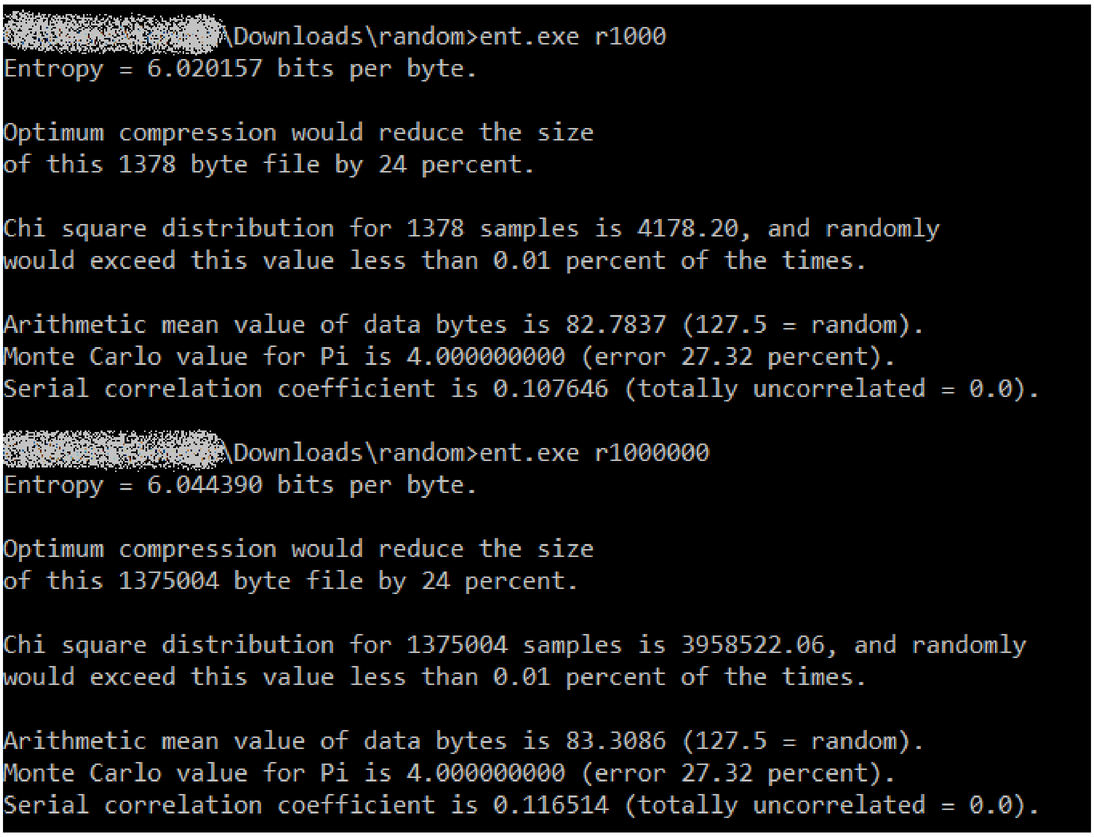
a)

⇒ *openssl rand [options] num: generates a random file of num bits.*

⇒ *Options: [-out r1000 -rand FILE] r1000 is the output file and the seed of the pseudorandom number generator is achieved from file FILE (or any ther file)*

With the same seed different results are generated.

b) The results of executing ENT over generated files are the following (note that used FILE has been CA.pl. If you choose other FILE results may be different, though but not substantially).



```
[redacted]\Downloads\random>ent.exe r1000
Entropy = 6.020157 bits per byte.

Optimum compression would reduce the size
of this 1378 byte file by 24 percent.

Chi square distribution for 1378 samples is 4178.20, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 82.7837 (127.5 = random).
Monte Carlo value for Pi is 4.000000000 (error 27.32 percent).
Serial correlation coefficient is 0.107646 (totally uncorrelated = 0.0).

[redacted]\Downloads\random>ent.exe r1000000
Entropy = 6.044390 bits per byte.

Optimum compression would reduce the size
of this 1375004 byte file by 24 percent.

Chi square distribution for 1375004 samples is 3958522.06, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 83.3086 (127.5 = random).
Monte Carlo value for Pi is 4.000000000 (error 27.32 percent).
Serial correlation coefficient is 0.116514 (totally uncorrelated = 0.0).
```

Not all tests are passed (e.g. compression is far from 0). Then, these files are not random, though openSSL is used to generate files with random content.

Exercise 3:

a) Compress the file ".doc" from exercise 1, calculate entropy and compare the result with that achieved in exercise 1.

b) Encrypt the ".doc" from exercise 1 with OpenSSL, using the following command

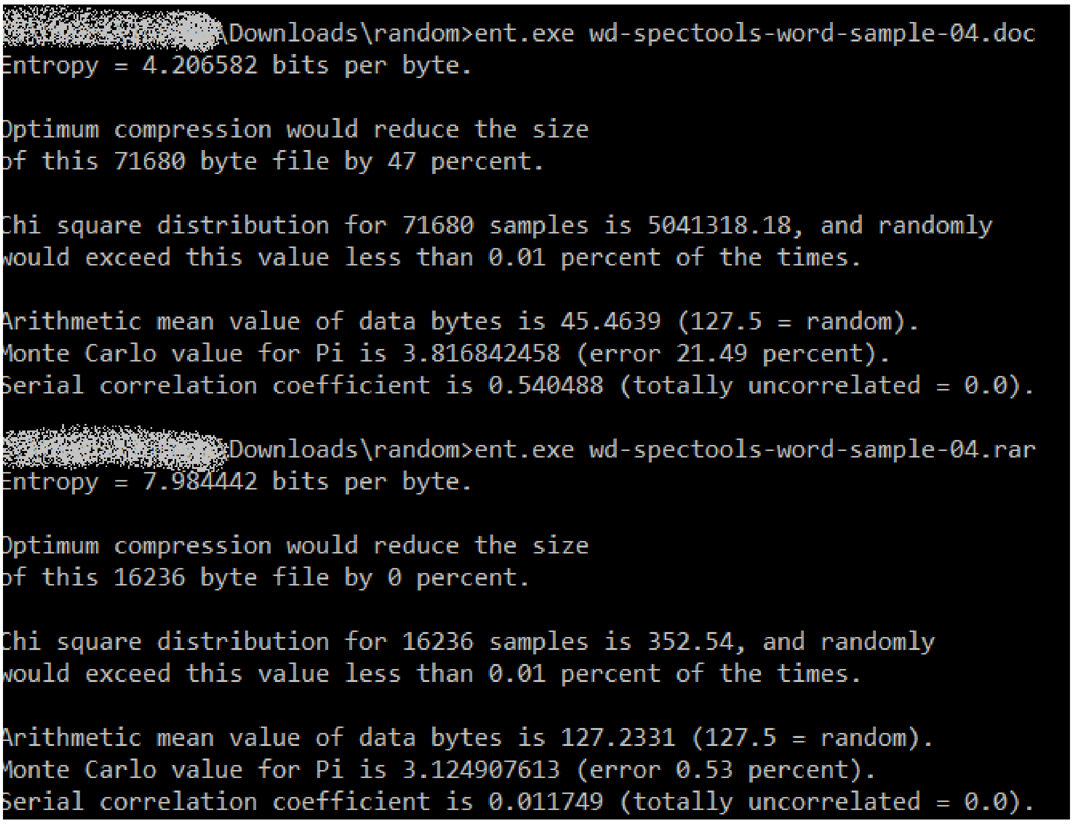
openssl enc -aes-256-cbc -salt -in FILE.doc -out FILE_ENCRYPTED.doc

Apply ENT over the resulting *"FILE_ENCRYPTED.enc" file and compare results with the entropy of the original file*.

c) Compress file *FILE.enc* with Winzip, Winrar or 7zip. Is the size affected? Explain why the size is or not affected. Compute entropy over this new file (compressed) and compare it with the original file and with the one generated in a) (encrypted).

Solution:

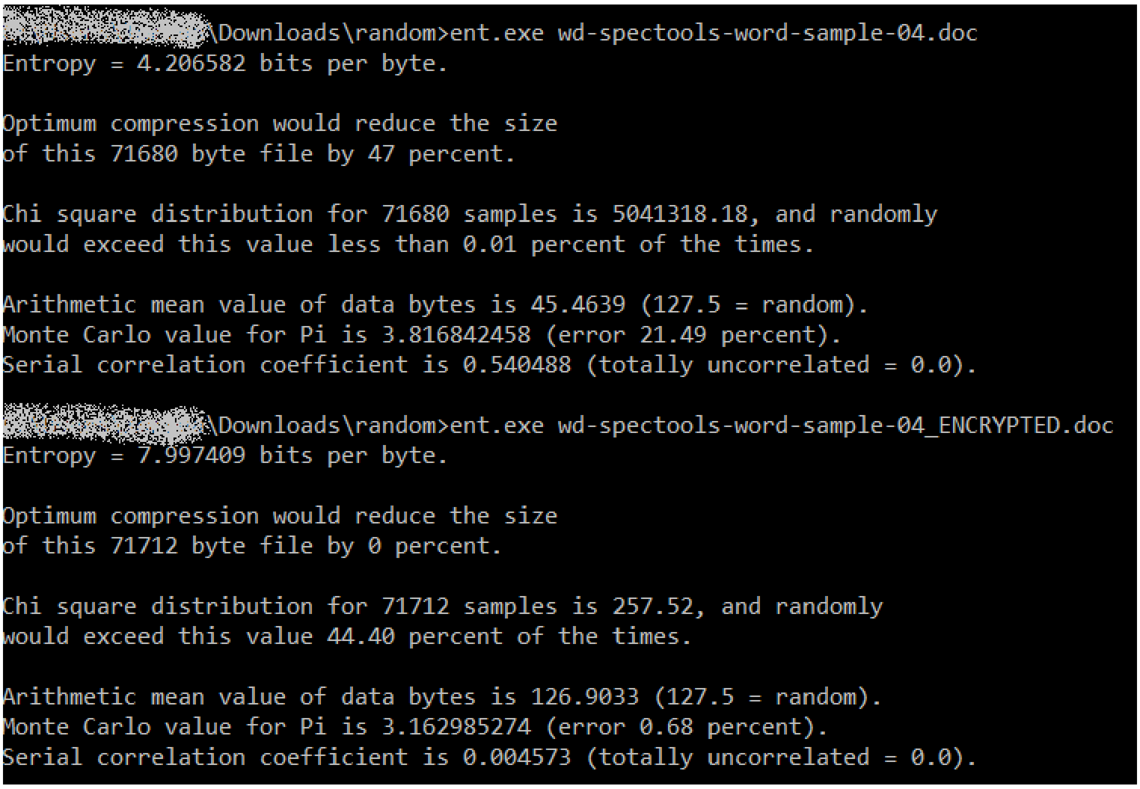
a) Results of executing ENT over the original .doc file and the compressed one are the following:



```
[redacted]\Downloads\random>ent.exe wd-spectools-word-sample-04.doc
Entropy = 4.206582 bits per byte.

Optimum compression would reduce the size
of this 71680 byte file by 47 percent.

Chi square distribution for 71680 samples is 5041318.18, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 45.4639 (127.5 = random).
Monte Carlo value for Pi is 3.816842458 (error 21.49 percent).
Serial correlation coefficient is 0.540488 (totally uncorrelated = 0.0).

[redacted]Downloads\random>ent.exe wd-spectools-word-sample-04.rar
Entropy = 7.984442 bits per byte.

Optimum compression would reduce the size
of this 16236 byte file by 0 percent.

Chi square distribution for 16236 samples is 352.54, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 127.2331 (127.5 = random).
Monte Carlo value for Pi is 3.124907613 (error 0.53 percent).
Serial correlation coefficient is 0.011749 (totally uncorrelated = 0.0).
```

Most tests are passed in the compressed file. Then, compression increases entropy. However, Chi square test is not passed (50 is the optimum) and then, none of the files is random.

b) The file has been encrypted with the word "Seguridad". Results of executing ENT over the original file and the encrypted one are the following:



```
[redacted]\Downloads\random>ent.exe wd-spectools-word-sample-04.doc
Entropy = 4.206582 bits per byte.

Optimum compression would reduce the size
of this 71680 byte file by 47 percent.

Chi square distribution for 71680 samples is 5041318.18, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 45.4639 (127.5 = random).
Monte Carlo value for Pi is 3.816842458 (error 21.49 percent).
Serial correlation coefficient is 0.540488 (totally uncorrelated = 0.0).

[redacted]\Downloads\random>ent.exe wd-spectools-word-sample-04_ENCRYPTED.doc
Entropy = 7.997409 bits per byte.

Optimum compression would reduce the size
of this 71712 byte file by 0 percent.

Chi square distribution for 71712 samples is 257.52, and randomly
would exceed this value 44.40 percent of the times.

Arithmetic mean value of data bytes is 126.9033 (127.5 = random).
Monte Carlo value for Pi is 3.162985274 (error 0.68 percent).
Serial correlation coefficient is 0.004573 (totally uncorrelated = 0.0).
```

Results of most tests have improved significantly when the file is encrypted. Given that Chi square test is a little bit far from the optimum value (50.00) and that the arithmetic mean is closed to 126.9 (though close to the expected value, 127.5), randomness is discarded. However, in this case randomness might be questionable.

c) When the file is compressed results are similar or worse, pointing out by Chi square test. This is linked to the fact that an encrypted file should have a high entropy and it may not be improved after compression.

Regarding file size, compression has not affected it because an encrypted file with high randomness, cannot be compressed. In fact, the size has increased a little due to data included by the compressor.