uc3m | Universidad Carlos III de Madrid

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?

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?

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).