COMP 7402 Assignment 5 Report

Peyman / Dimitry

Goal

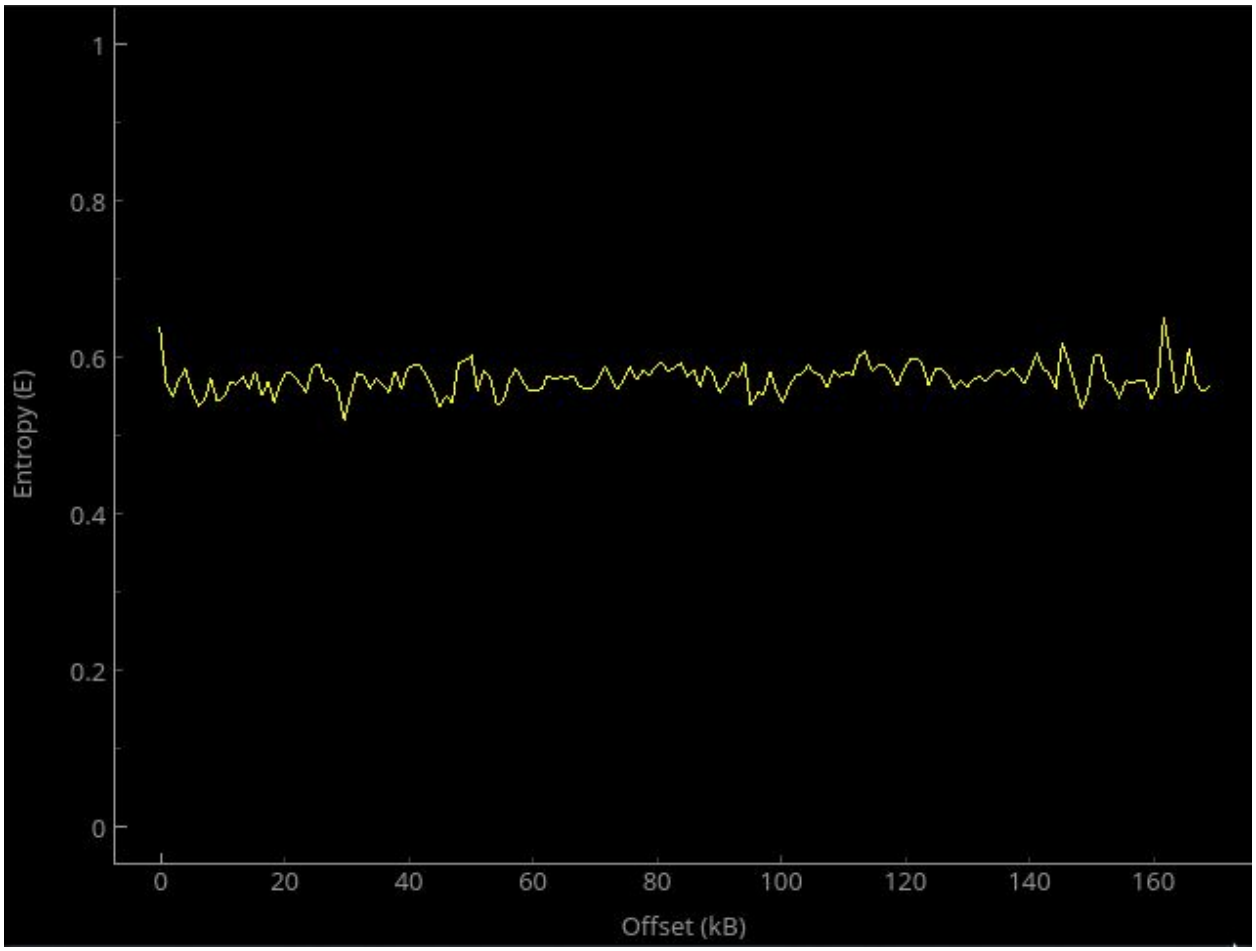
For this assignment our goal was to create a simple Feistel cipher with 8 rounds. The cipher would provide an option to use either ECB or CBC encryption schemes. Another feature that would be added to the previous assignment would be sub-key generation.

Analysis

This implementation causes the ciphertexts to be rather random. During testing we found that to the naked eye the output of the ECB function appeared less random than the output from the CBC function. This had to be confirmed using binwalk's entropy setting.

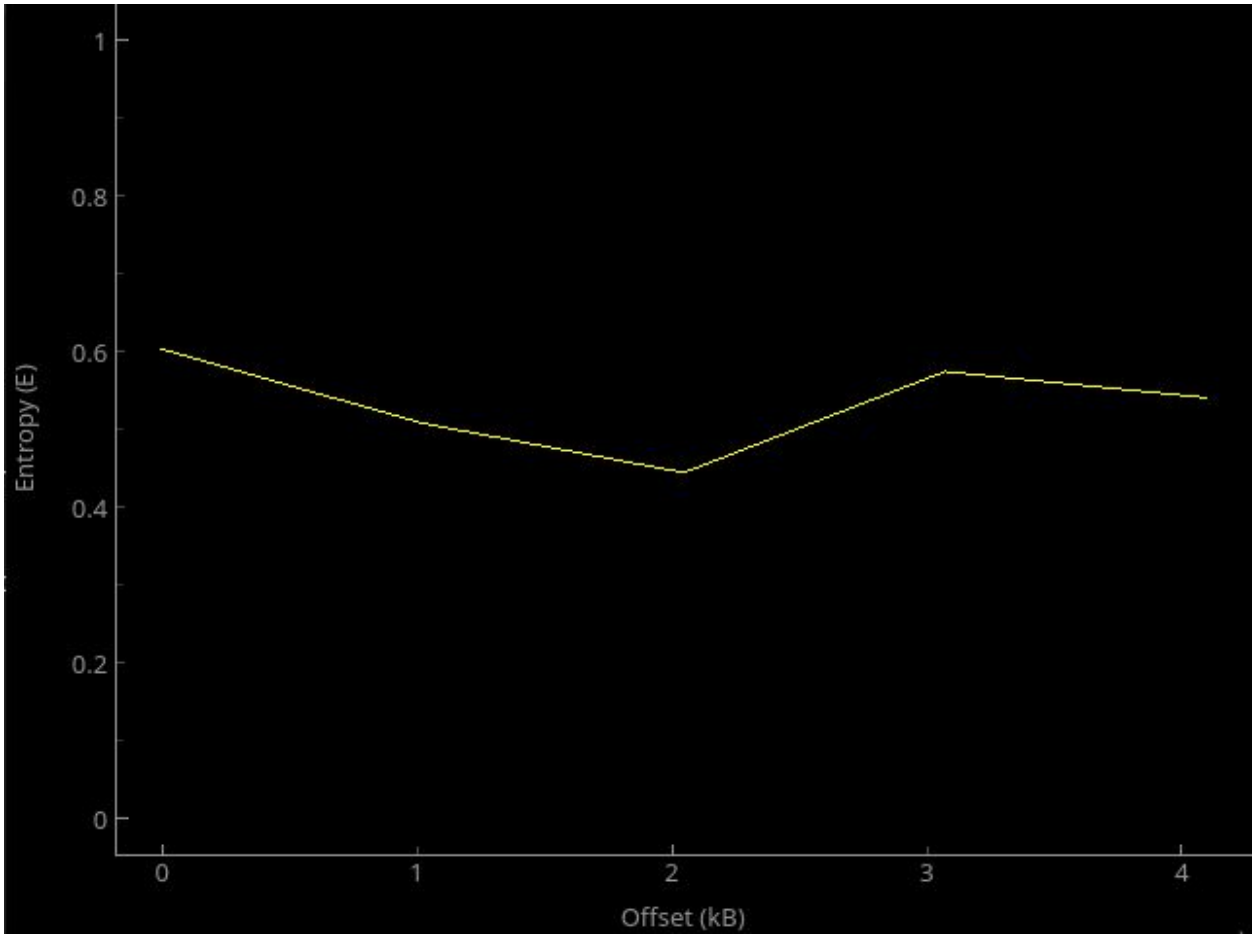
Alice Entropy

First it would help to take a look at the entropy of the english language. Taking a rather large sample such as "Alice in Wonderland" we can see that the entropy is around 0.6.



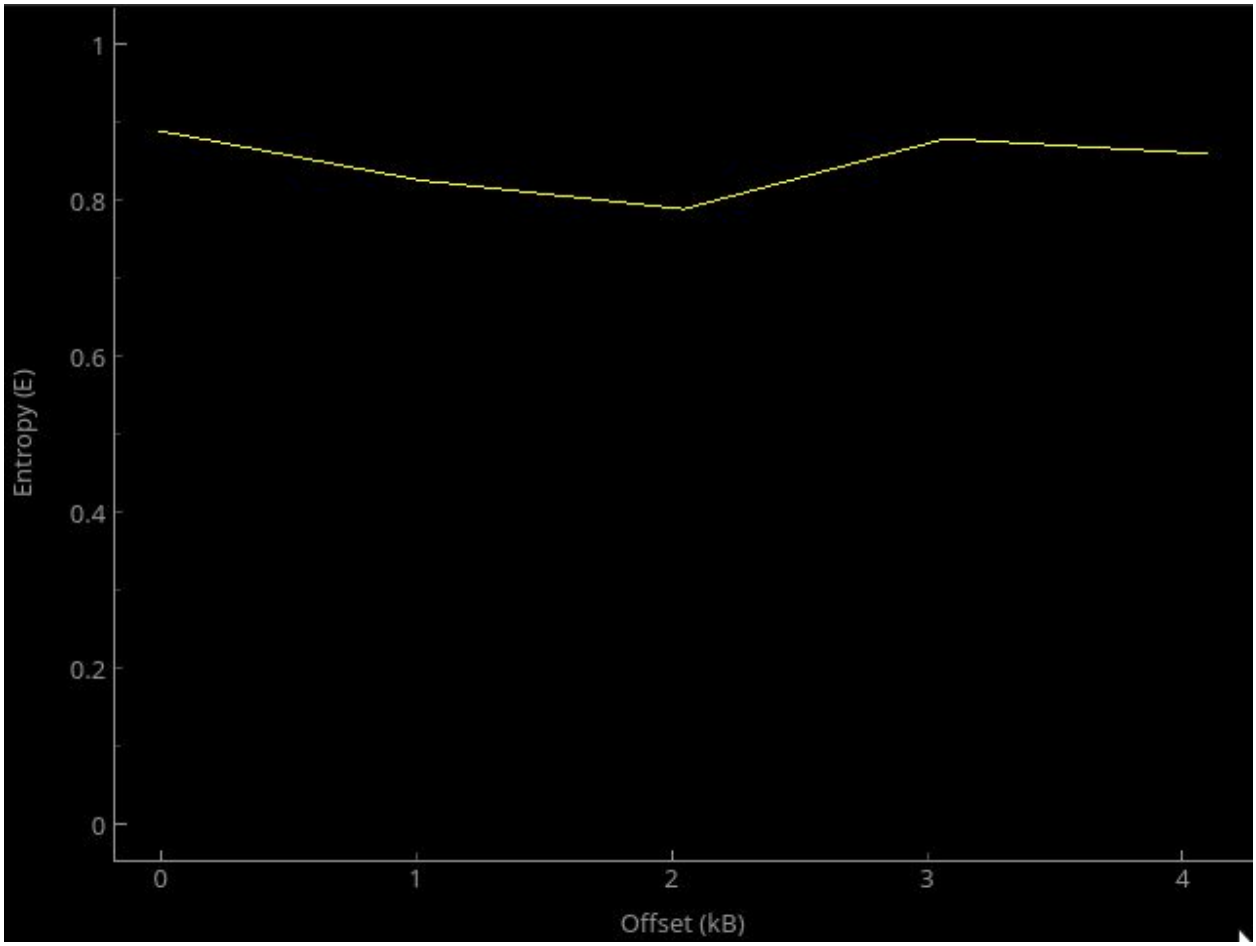
Source Code Entropy

Following this we can attempt to analyze the entropy of out plaintext file src/feistel.c. This file comes out to have a roughly similar entropy to the large text file we tested above.



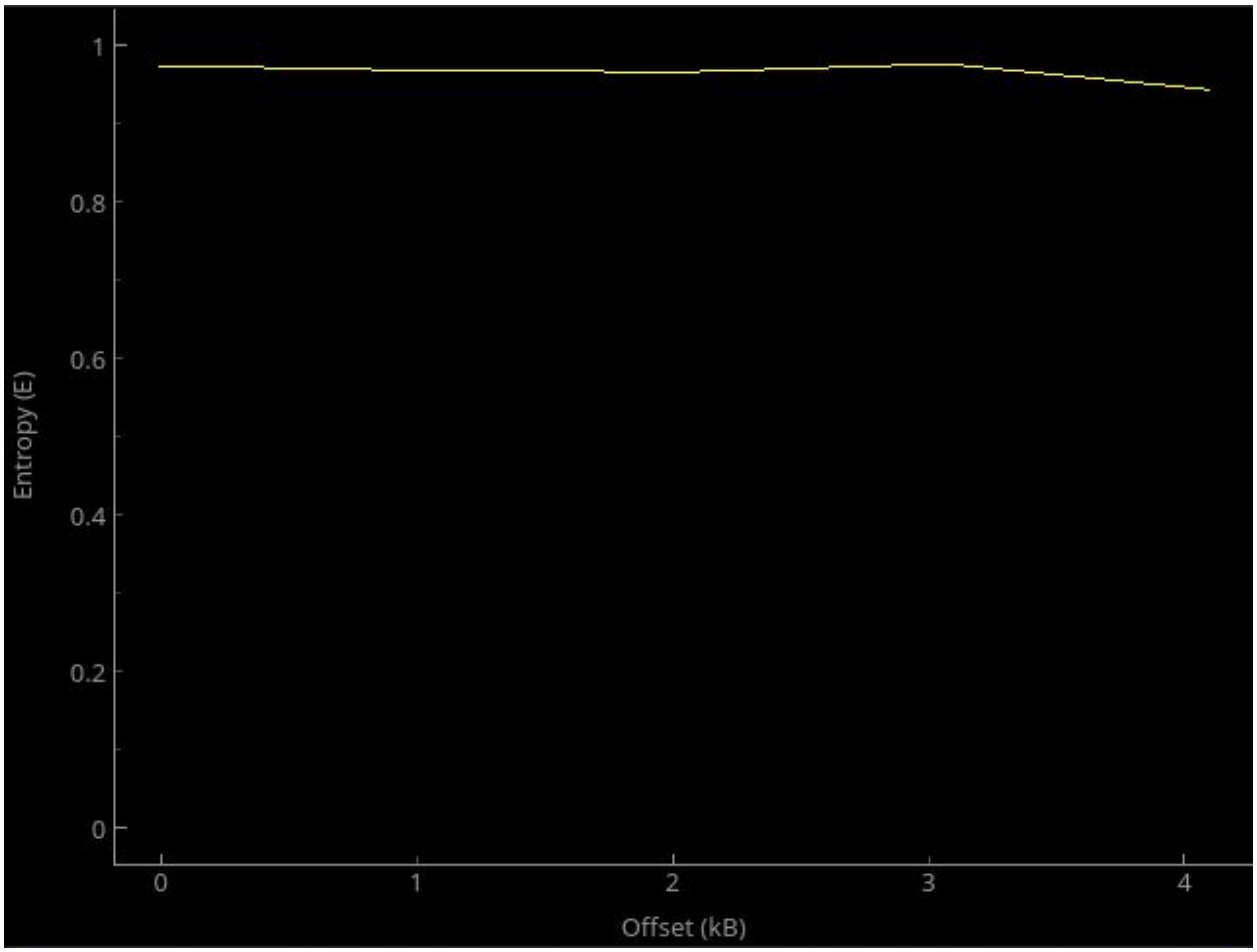
ECB Encrypted Source Code Entropy

Using this new found information we tested the output file from the ECB function and found that the entropy is much higher than in a plaintext file. This shows that the diffusion of data is much greater because the randomness of the bits has increased greatly.



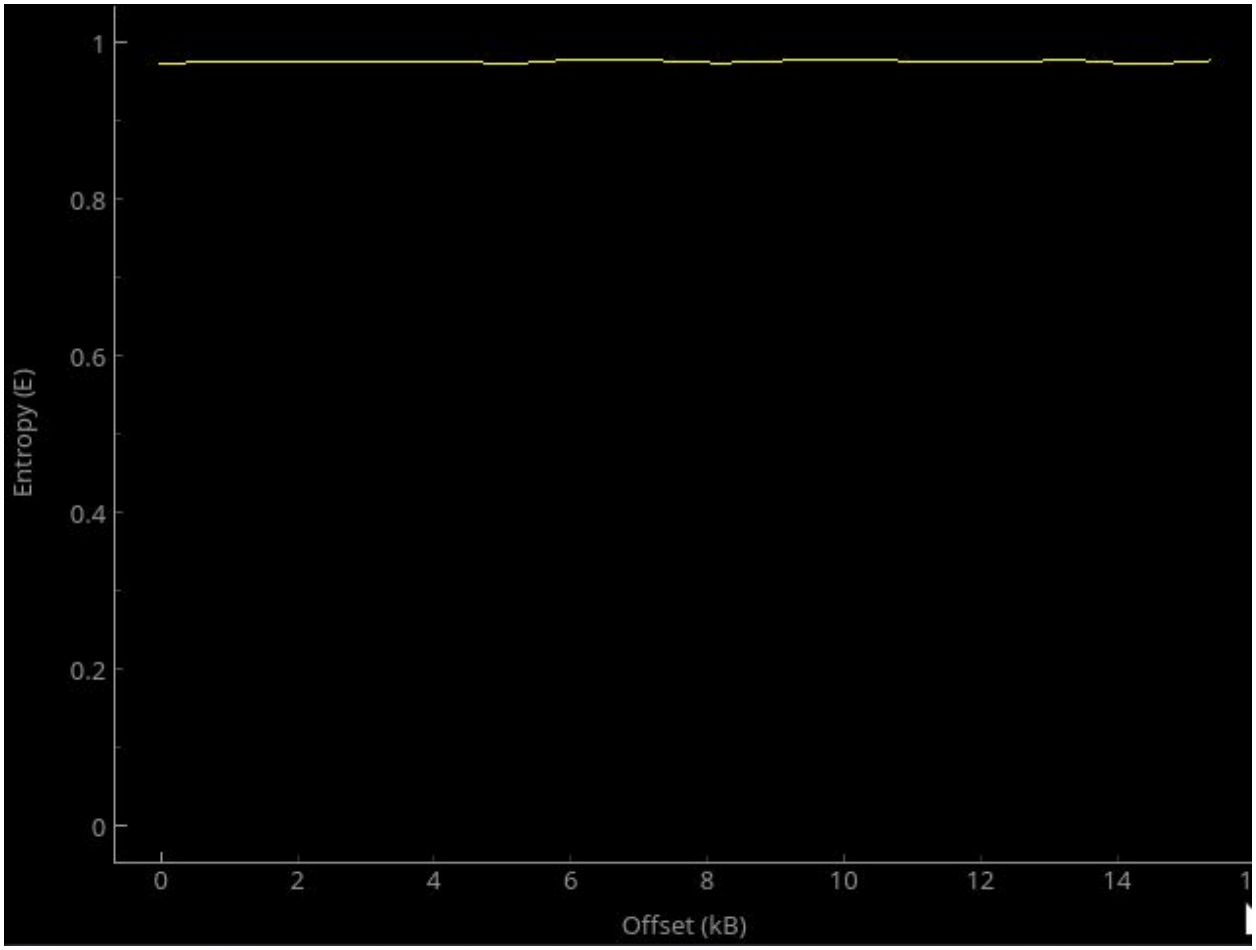
CBC Encrypted Source Code Entropy

The results of this test were rather unexpected. The entropy of the same file encrypted using CBC was much higher than that of ECB. In face the entropy was almost 1 which was rather impressive. One could almost not distinguish the data from a block of data generated from /dev/urandom in the next section.



Entropy of Data From /dev/urandom

The entropy of urandom was checked to see how high the entropy is. Here we see that while it is much more stable in the range of ~1 it is still very similar to the output of the CBC mode.



Conclusion

Due to the evidence collected it appears that CBC mode is much more secure than ECB mode. This is probably the case because in CBC mode the previous block affects the encryption of the each block following it. Since they are passed through xor for each block, they end up much more random than ECB which only relies on the changes done during the permutation stage.