Parallel computing

Logbook - Adam Stock

University of the West of England

UFCFFL-15-M - Parallel computing

# First attempts 19/11/18

## Initial attempt 1.c

This initial version was created to explore the code given to us for reading and contains little in the way of student code, however reading it provides some insight into the workings of the encryption and decryption functions.

## Version 2.c

This version was created as a framework for future usage, it contains #includes for most things I could think of at the time that may be required based on work completed in previous lab sessions.   
the code itself has listing for items that will be used in all future version of the program.

These are as follows –

char truekey [16] = "#####123456#####";

This is where that actual key we are trying to find will be stored, note this has been artificially  
limited by way of padding with ‘#’ and the usage of only numerical characters.

I have used 6 characters each containing 0-9 giving us 10^6 or 1 million possibilities – this could easily be expanded but the computational power required for this would be beyond the scope of this assignment.

char originaltxt [16] = "thisisatestxxxxx";

This is the text that will be encrypted by the AES function its actual contents matters little but must remain consistent in order to provide useful comparison.

char encryptedtxt [16];

This is the same text as above but after the AES function has been completed.

char work [16];

This will be where the candidate encrypted text will be stored while awaiting comparison to the encrypted text mentioned above.

char testkey [16] = "#####xxxxxx#####";

This is where the working key to be tested will be stored with ‘x’ as a placeholder to be generated later.

The ‘int main’ contains a basic nested for loop to generate the desired test key as well as some commented pseudo-code outlining further plans to continue this.

The function is designed to print out each test key sequentially so that its accuracy can be checked.