Lab #2 by William Lentini, Shamar Roberts, and Daniel Lake

1. Explain the full steps you would take in encrypting messages to your partner. You can explain in terms of the Kryptos steps we used before. No need to use screenshots in here, just explain your train of thought.
    a. In order to encrypt the message to my partner our first step was changing our Kryptos to sender. Then we chose the operation to encrypt and the Algorithm of Rijndael.After that we set our parameters to default, generated a key, and set the mode to ECB.We then entered the file we wanted to encrypt into the input file and chose a name that the file would be once it was encrypted. After that we pressed go and the message was encrypted

2. Explain the full steps that your partner would take in order to decrypt the message you sent him. The receiving partner will have to answer this part, and it should be compiled in the final report. I need to know exactly what the receiving partner had to do to "understand" the message you have sent him.
    a. To understand the message that I was sent I first changed my status to receiver and changed my Operation to decrypt. Then I set my algorithm to Rijndael and my parameters to default. After that I entered in the private key and switched mode to ECB and input the Roberts_Shamar_AES_ECB128.exe file and provided the name for the output file as Roberts_Shamar_decrypted. Once that was done I hit Go and the message was decrypted.

3. If you are the sender what information aside from the encrypted message did you have to communicate to your partner?
    a. Aside from the encrypted message I had to communicate the correct key symmetric utilizes one key.

4. If you are the recipient what information did your partner communicate to you in order to decrypt the message?
    a. If I'm the recipient my partner would have to communicate with me the correct key in order for me to decrypt the message.

5. How was the other information (steps 3 and 4 above) (not the encrypted message) obtained from/to the partner?
    a. You could receive that information through email.

6. Was it a good (secure) way to obtain the information? Explain and elaborate!

    a. Email is not a secure way to obtain the information, there are plenty of times where people’s emails get leaked or hacked into. If it does get hacked they would have access to the key.

7. What might happen if the information you shared on steps 3 and 4 with your partner was intercepted? Explain and elaborate!
    a. If a person intercepts the message this would be bad because with symmetric encryption anyone with the secret key can decrypt the information.

8. Did you notice any major changes in performance when you encrypted the small-plaintext file or the large file? Put in here any observations here with respect to how the file size affects the symmetric encryption/decryption operations.
    a. I noticed that with a larger file size there is a major change in CPU time and Ticks. For the small-plain text file with I/O, I noticed the CPU time was 0.000310 seconds and the amount of ticks were 989,613 cycles. But for the large doi.file timing with I/O, the Cpu time was 2.212172 seconds and the CPU ticks were 7,061,262,236 cycles. This just goes to show the bigger the file there’s more of a process that goes to encrypting. When decrypting the files it’s basically the same when decrypting the doi file the CPU time and CPU Ticks were nearly the same as when I was encrypting the file. Large files take more processing than smaller files

9. Extra credit: Be brave if you wish, and try to use the symmetric encryption of Kryptos to encrypt a very large file which is located here. This is a nearly 1.2GB file. How do your measurements change when you encrypt/decrypt this file?

I did try to download the 1.2GB file but for some reason it was not working but I copied and pasted some of the document and still tried messing around with it to see what kind of results I would get. The file was larger than the small plain text file but still smaller compared to the large doi file. What I witnessed confirmed my beliefs that larger files take more processing time. For the part of the declaration of Independence that I did encrypt I noticed The timing with I/O for Cpu time was 0.085947 seconds and the CPU ticks were 274,342,3222 cycles. That is a shorter amount of time compared to the doi file which Cpu time was 2.212172 seconds and the CPU ticks were 7,061,262,236 cycles. This makes sense, the declaration file I had was smaller than the Doi file. I believe if I had been able to download the Declaration of Independence file, it would take way more time than it had taken for the DOI file.

Reflection Questions

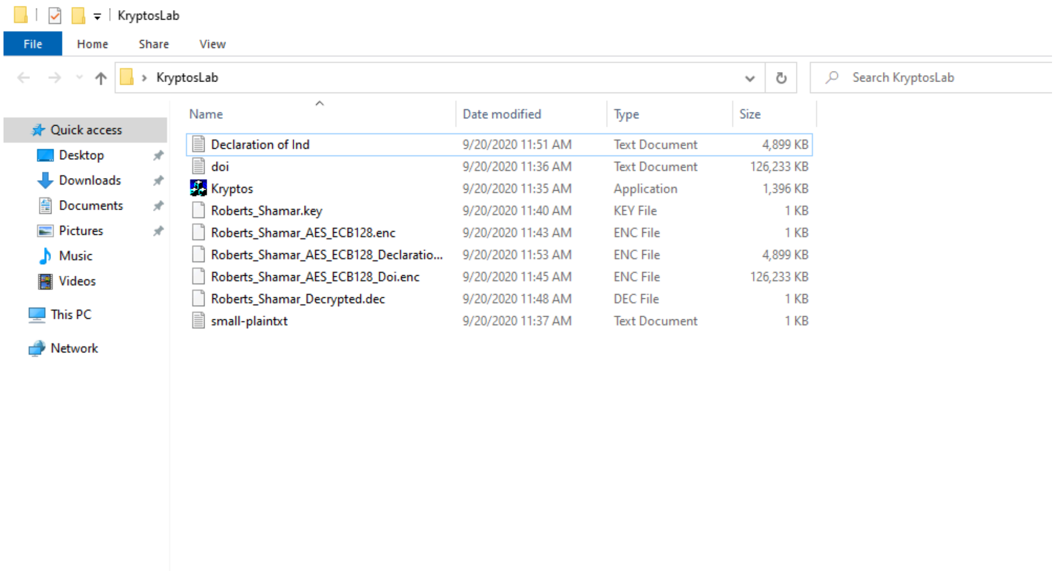
1. What was the most challenging part of this activity?

    a. The most challenging part of this activity was figuring out how to see the encrypted message that I (Will Lentini) sent to Shamar Roberts

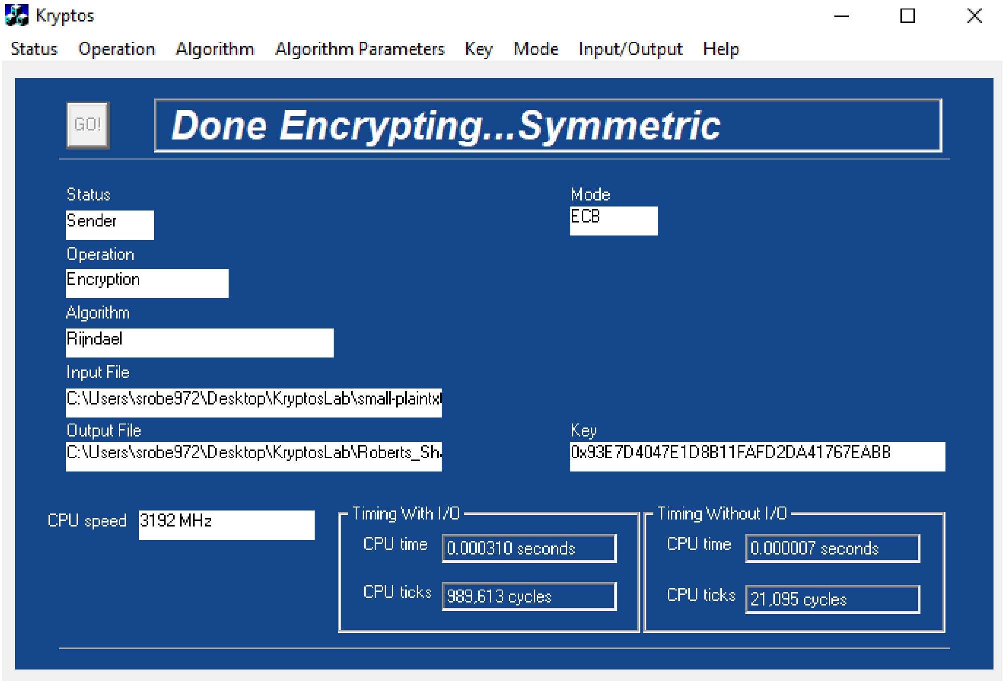
2. What was the most enjoyable part of this activity?
    a. The most enjoyable part of the lab was filling in the information for the encryption and decryption process.

3. Do you think you have a slightly better understanding of how a symmetric key works i.e., what are some symmetric key encryption algorithms you can use as well as their setting? Why or why not?
    a. I think that we have a better understanding of how symmetric keys work because we learned how to send an encrypted message to one of our group members and we encrypted it successfully. Some symmetric key encryption algorithms we can use are block and stream algorithms to encrypt data. The first algorithm, block data, is when you encrypt data block by block. A block is a specified set of bits that are encrypted using a secret key. The only drawback with the block algorithm is that after you encrypt something, the encrypted data is retained in the memory components, and this retention of data is done when the system actually waits for complete blocks of data. The wait time can lead to security gaps that can compromise the security of the data. The second algorithm, stream algorithm, is when data is encrypted bit by bit, making it the faster option . With this option, the data is not retained in the memory of the system, making it arguably a safer option. Only drawback about this type of algorithm is that is it more difficult to implement properly than a block algorithm.

4. What would you like to change from this activity?
    a. What we would like to change from this activity is the vagueness of some of the questions. I understand that the questions are meant to be thought about hard, but if some groups think and interpret the questions in the incorrect way, they potentially have the chance to answer it in a way that you as the professor were not looking for and that would result in the students getting the question wrong.



Name	Date modified	Type	Size
Declaration of Ind	9/20/2020 11:51 AM	Text Document	4,899 KB
doi	9/20/2020 11:36 AM	Text Document	126,233 KB
Kryptos	9/20/2020 11:35 AM	Application	1,396 KB
Roberts_Shamar.key	9/20/2020 11:40 AM	KEY File	1 KB
Roberts_Shamar_AES_ECB128.enc	9/20/2020 11:43 AM	ENC File	1 KB
Roberts_Shamar_AES_ECB128_Declaratio...	9/20/2020 11:53 AM	ENC File	4,899 KB
Roberts_Shamar_AES_ECB128_Doi.enc	9/20/2020 11:45 AM	ENC File	126,233 KB
Roberts_Shamar_Decrypted.dec	9/20/2020 11:48 AM	DEC File	1 KB
small-plaintxt	9/20/2020 11:37 AM	Text Document	1 KB



Done Encrypting...Symmetric

Status: Sender

Mode: ECB

Operation: Encryption

Algorithm: Rijndael

Input File: C:\Users\srobe972\Desktop\KryptosLab\small-plaintxt

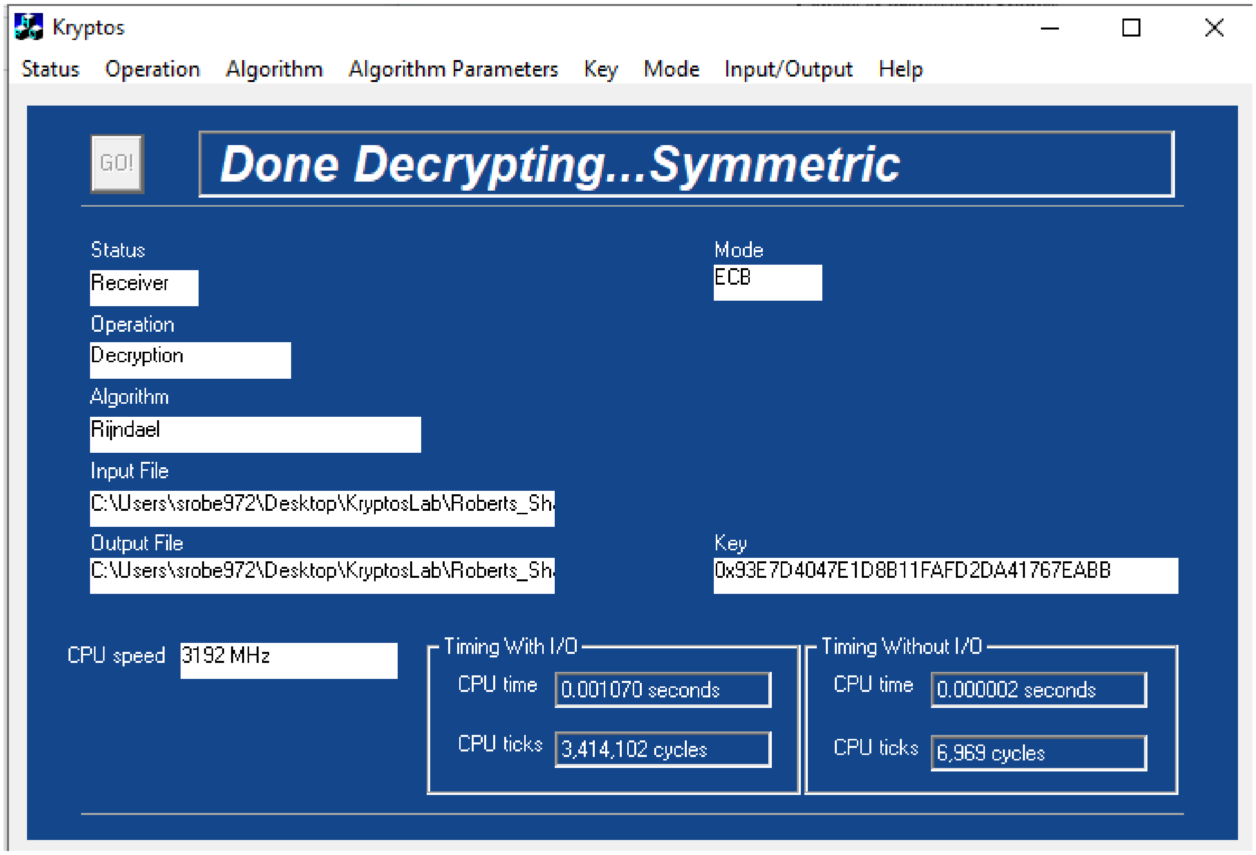
Output File: C:\Users\srobe972\Desktop\KryptosLab\Roberts_Sh

Key: 0x93E7D4047E1D8B11FAFD2DA41767EABB

CPU speed: 3192 MHz

Timing With I/O
CPU time: 0.000310 seconds
CPU ticks: 989,613 cycles

Timing Without I/O
CPU time: 0.000007 seconds
CPU ticks: 21,095 cycles


Kryptos
Status Operation Algorithm Algorithm Parameters Key Mode Input/Output Help
GO!

Done Decrypting...Symmetric

Status: Receiver

Operation: Decryption

Algorithm: Rijndael

Input File: C:\Users\srobe972\Desktop\KryptosLab\Roberts_Sh

Output File: C:\Users\srobe972\Desktop\KryptosLab\Roberts_Sh

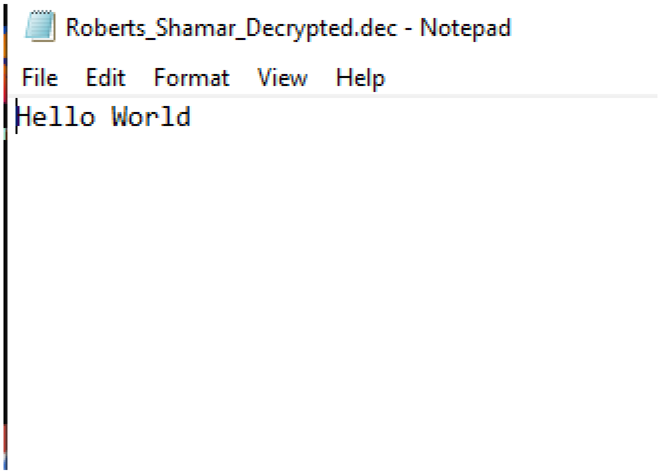
Mode: ECB

Key: 0x93E7D4047E1D8B11FAFD2DA41767EABB

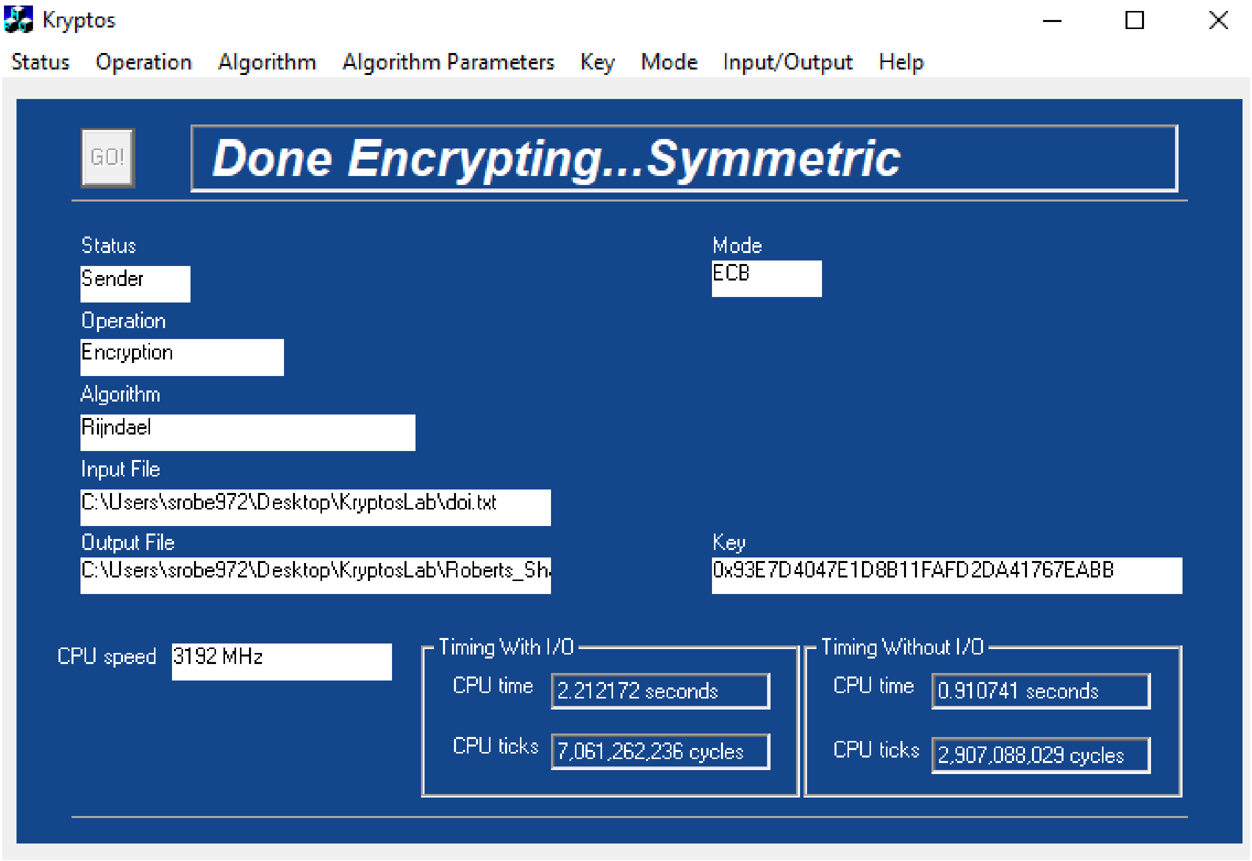
CPU speed: 3192 MHz

Timing With I/O
CPU time: 0.001070 seconds
CPU ticks: 3,414,102 cycles

Timing Without I/O
CPU time: 0.000002 seconds
CPU ticks: 6,969 cycles


Roberts_Shamar_Decrypted.dec - Notepad
File Edit Format View Help

```
Hello World
```



Done Encrypting...Symmetric

Status
Sender

Operation
Encryption

Algorithm
Rijndael

Input File
C:\Users\srobe972\Desktop\KryptosLab\doi.txt

Output File
C:\Users\srobe972\Desktop\KryptosLab\Roberts_Sh

Mode
ECB

Key
0x93E7D4047E1D8B11FAFD2DA41767EABB

CPU speed 3192 MHz

Timing With I/O
CPU time 2.212172 seconds
CPU ticks 7,061,262,236 cycles

Timing Without I/O
CPU time 0.910741 seconds
CPU ticks 2,907,088,029 cycles