CBC & CTR FEISTEL CIPHER

Sukh Atwal, Yoshi Ryuzaki

Table of Contents

Introduction

The Feistel cipher isn't so much a cipher in itself, it's more so of a framework for building encryption algorithm. It's widely used in modern ciphers and has various different modes that affects both the confusion and diffusion properties.

Building off of the previous Assignment #5, we are going to add onto ECB mode with CBC and CTR. In addition to that, we are also going to implement a basic permutation scheme to generate subkeys from the main key that we input.

User Guide

Here is the user guide of the application. Note that the functionality was verified and tested using Python 3.8 under a Windows environment. An README.md file will also be included with the same instructions located in the same folder.

When you first open the 'Source' folder, there should be 5 python script files and a few sample text/image files.

decryption.py	12/1/2020 1:00 AM	Python File	9 KB
encryption.py	12/1/2020 12:44 AM	Python File	10 KB
filehandling.py	11/11/2020 1:45 PM	Python File	1 KB
main.py	12/1/2020 1:03 AM	Python File	4 KB
sample.txt	10/21/2020 10:02 PM	Text Document	4 KB
sample1.BMP	11/8/2020 6:33 PM	BMP File	770 KB
sample2.bmp	11/9/2020 10:24 PM	BMP File	3,199 KB
subkeygen.py	12/1/2020 12:55 AM	Python File	3 KB

Encryption

Here is the step-by-step guide for the encryption process:

1. First, we will launch the application using 'python main.py' or 'python3 main.py' command:

```
C:\Users\yryuz\Documents\BCIT\Crypto\Assignment6\final>python main.py
```

2. Next, after getting prompt, we will have to enter the operation mode we want to use (either encryption or decryption):

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
:1
```

3. For now, we will use operation mode 1 which is the encryption mode. And the application will prompt us to choose the file type.
   We will select 1 (Image File type) here:

```
:1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
:1
```

4. After selecting using what type of file, the application will prompt us for input file name. And we will use sample1.bmp for this case:

```
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
:1
Please enter the input file name/path: sample1.bmp
```

5. Next, we will also have to enter the output file name where we want to keep the encrypted data:

```
Please enter the output file name/path: encrypted.bmp
```

6. And now, we will create and enter a key phrase which we want to use to encrypt the file. The key phrase can be either alphabets or numbers. And the application only accepts the password which is longer than 3 characters. (The application also returns all the subkeys which will be generated based on the user entered key phrase)

```
Please enter the encyrption key to use(Can be Alphabet or Number. For best security, more than 3 character long):hello
Generate Subkey:141
Generate Subkey:127
Generate Subkey:247
Generate Subkey:184
Generate Subkey:29
Generate Subkey:231
Generate Subkey:250
Generate Subkey:222
```

7. The application also prompts us to choose the encryption mode.
(This time we will choose CBC. When choosing CTR, user will also have to enter the initial counter value)

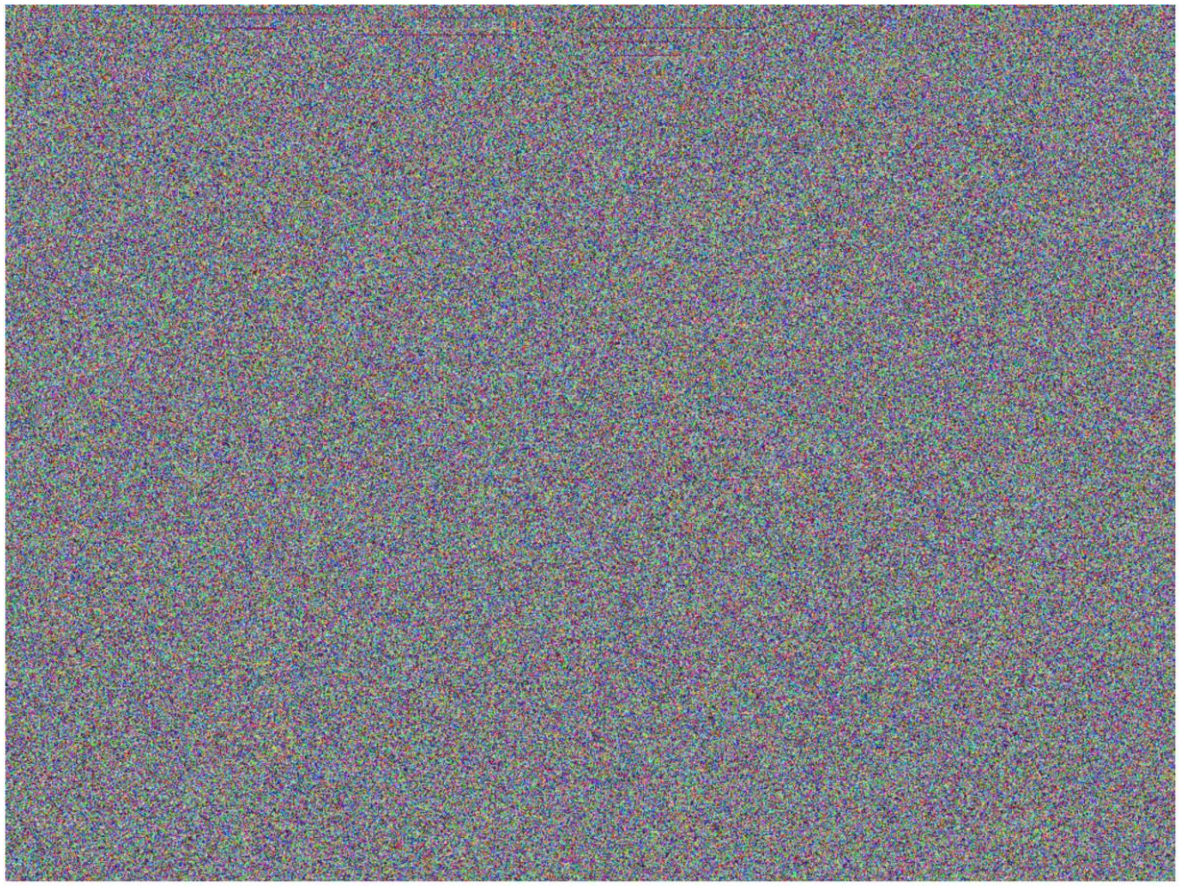
```
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
:2
```

8. Then the application returns ‘Encrypting, please wait’, and we will wait till the application prints ‘Finished Encryption’ message.

```
-------------------------------------------------------------------------------------------------------
 Encrypting, please wait
-------------------------------------------------------------------------------------------------------
Skipping header: 54bytes long
```

9. Finally, the encryption process is over, you should find a file called encrypted1.bmp.

```
Finishsed encryption, please check the folder
```



Decryption

The decryption process is very similar to the encryption process.
Here is the step-to-step guide of the decryption process:

1. First, we will use 'python main.py' or 'python3 main.py' to launch the application:

```
C:\Users\yryuz\Documents\BCIT\Crypto\Assignment6\final>python main.py
```

2. Next, we will be choosing the operation mode. Since we want to decrypt the file this time, we will be choosing 2:

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
:2
```

3. The application then prompts us to enter the file type, we will pick 1 this time:

```
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
:1
```

4. And we also have to pass the input and output fie names/paths to the application when getting prompted:

```
Please enter the input file name/path: encrypted1.bmp
Please enter the output file name/path: decrypted1.bmp
```

5. Next, we will enter the key phrase we used for encryption:

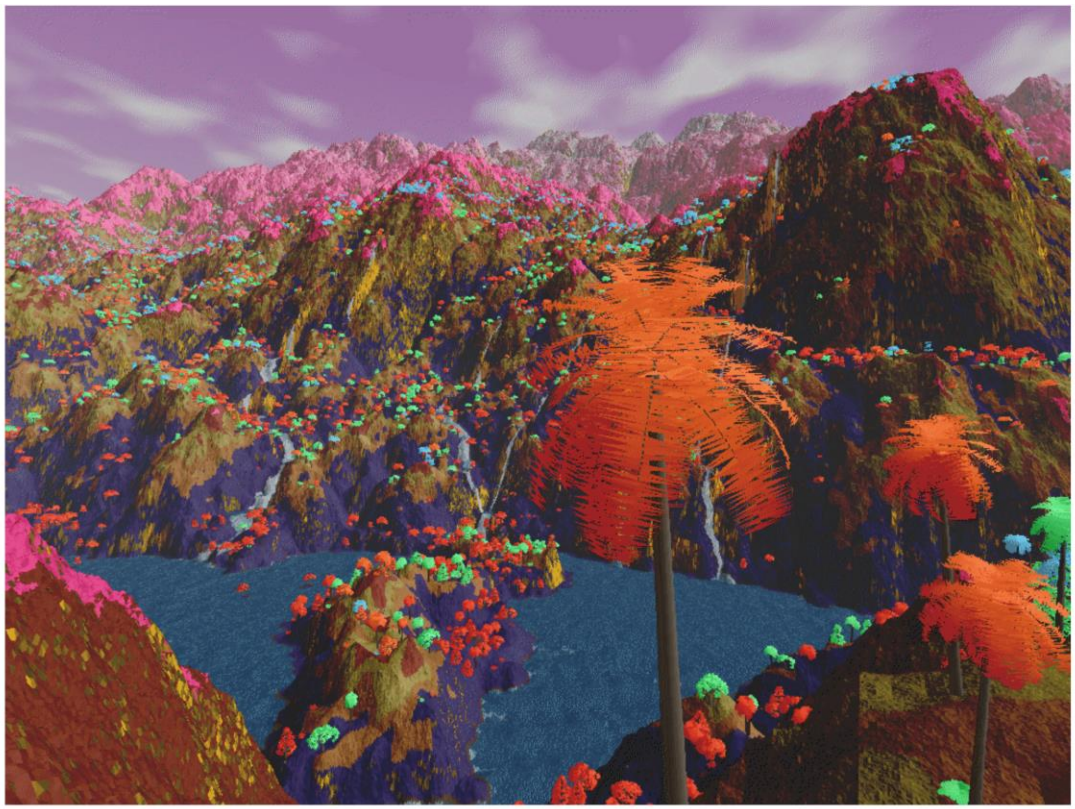
```
Please enter the encyrption key to use(Can be Alphabet or Number. For best security, more than 3 character long):hello
Generate Subkey:141
Generate Subkey:127
Generate Subkey:247
Generate Subkey:184
Generate Subkey:29
Generate Subkey:231
Generate Subkey:250
Generate Subkey:222
```

6. And at last, we will choose the encryption mode we used for encrypting the file, and just wait for the decryption process to be finished:

```
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
:2
-----------------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------------
Skipping header: 54bytes long
```

7. Once the application returns ‘Finished decryption’ message, then we can check the folder and find the decrypted1.bmp file.

```
Finishsed decryption, please check the folder
```



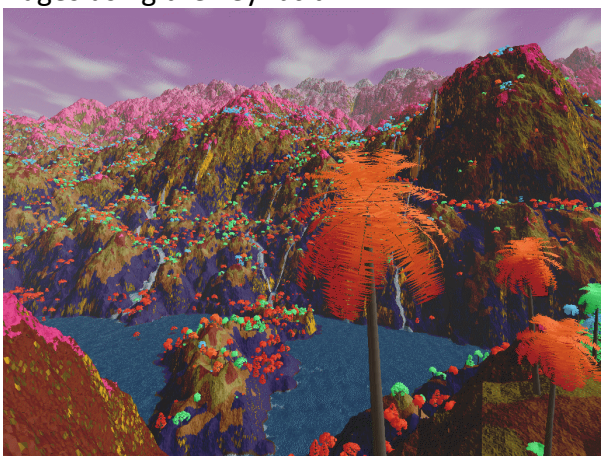
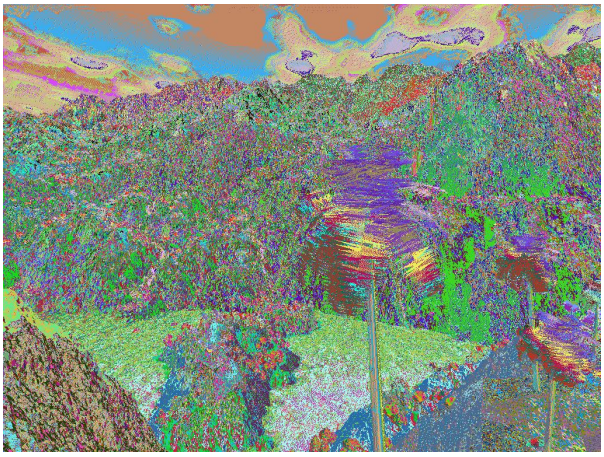
Analysis

ECB (Electronic Codebook)

ECB otherwise known as Electronic Codebook is one of the easier modes, where the plaintext is divided into pre-determined block sizes and then performs encryptions on them using the provided key. To decrypt, it's the same process.

Diffusion

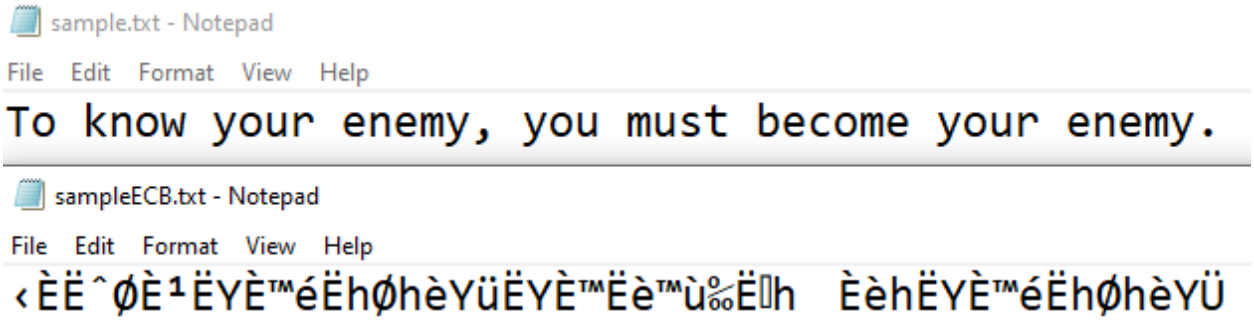
One of the biggest issues with ECB is diffusion, more specifically the lack of it. Take a look at the following two images using the key 'bcit'.

The image on the right is the original image and the one on the left is after encrypting it. We can see parts of the image match with one another, showing that ECB doesn't hide the pattern of the data very well.
Each individual pixel might be encrypted but the overall image can still be perceived since the pattern of colors in the original still remain in the encrypted. We can still see the general colors from the image on the left in the image on the right. This is more of a problem with the way the encryption mode, and there are better modes to use such as CBC or CTR which we'll look at.

Confusion

When looking at the confusion side, I'm looking at the relationship between the ciphertext and key and whether it's possible to get the key from the ciphertext.



Here I used the same key as the image 'bcit', and it produces essentially random text, there's no way to get the key. An issue however is that if two messages and the key are the same, the ciphertext would output the same both times. However, there's no way you can work back the key or encryption algorithm from just the ciphertext alone. Therefore, I would say that this algorithm has moderate confusion. Moderate because while implementing using a Feistel structure does improve the confusion, ECB is a bottleneck in terms of how much you can do with it. To improve the confusion, we could increase the block size or use other block mode ciphers as mentioned before such as CBC.

Avalanche Effect

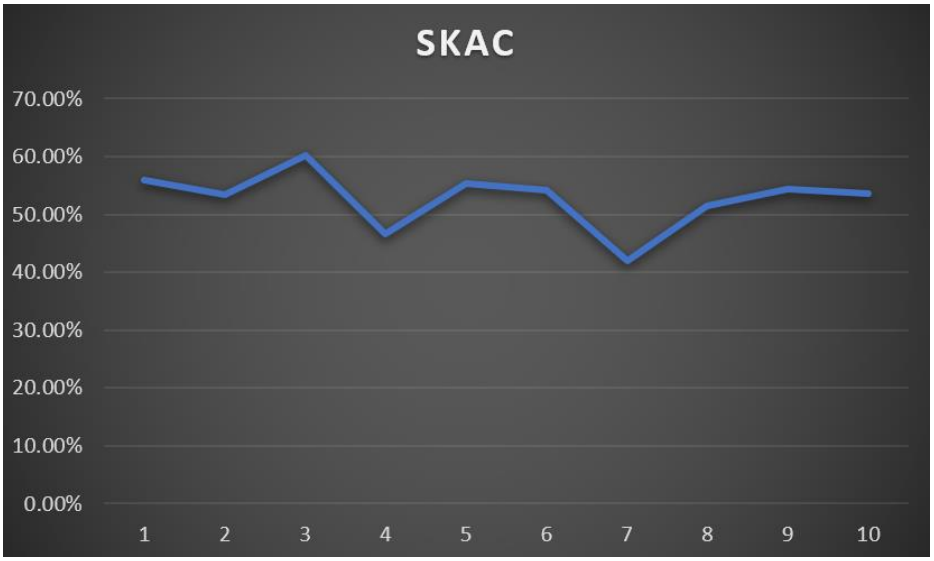
The following is a table of bits changed in the plaintext as well as changes in the key for SPAC.

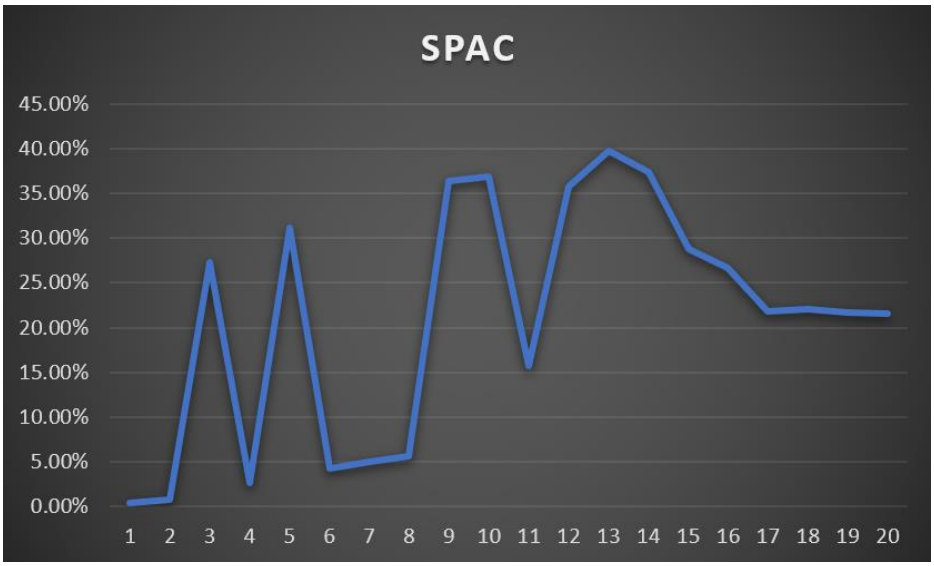
Round	Bits Changed in Cipher	Percentage Changed
1	3	0.40%
2	6	0.80%
3	204	27.35%
4	20	2.65%
5	232	31.10%
6	32	4.24%
7	37	4.96%
8	42	5.63%
9	269	36.45%
10	272	36.86%
11	117	15.66%
12	264	35.77%
13	290	39.78%
14	276	37.40%
15	215	28.78%
16	202	26.72%
17	163	21.82%
18	165	22.09%
19	162	21.69%
20	161	21.55%

As well as for SKAC.

Key	Bits Changed in Cipher	Percentage Changed
1	528	55.87%
2	471	53.40%
3	628	60.15%
4	377	46.54%
5	519	55.45%
6	346	54.15%
7	310	42.01%
8	431	51.49%
9	524	54.41%
10	487	53.58%

To get a clearer understanding, I've converted them to line graphs.





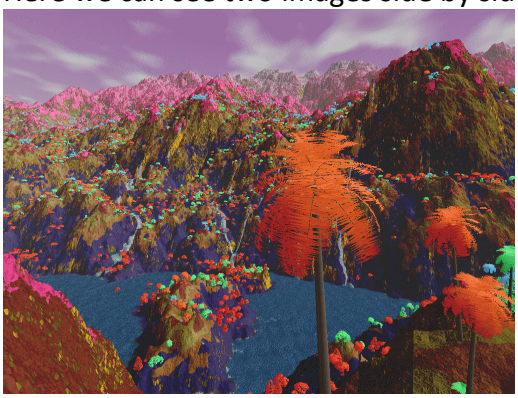
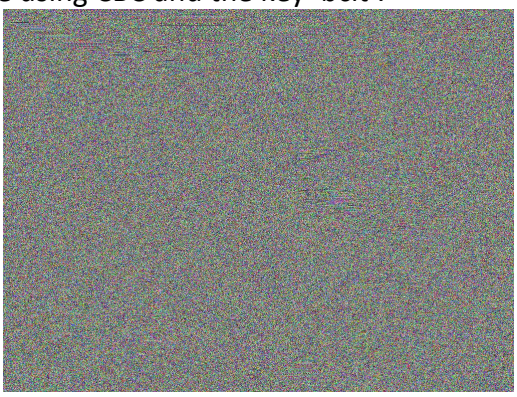
SKAC data was collected using the same plaintext but different keys (From 1-10). Meanwhile, SPAC data is from modifying plaintexts one at a time. It's hard to say whether the change percentage would remain stagnant if we went beyond round 20 for SPAC however it's still on the lower side. Compared to SKAC, which is consistently hitting 50% and only dropped to 40% in the number of bits changed once. I would say this is a somewhat desirable area to hit however there are still rooms for improvement.

CBC (Cipher Block Chaining)

CBC otherwise known as Cipher Block Chaining is, in a sense, an extension of ECB mode. The plaintext is divided into blocks and then encrypted with the key. Before the encryption is actually performed, the plaintext block is XOR'd with the ciphertext of the previous block which ends up increasing the confusion and diffusion properties, a weakness of ECB that was pointed out previously. To decrypt, the process is reversed.

Diffusion

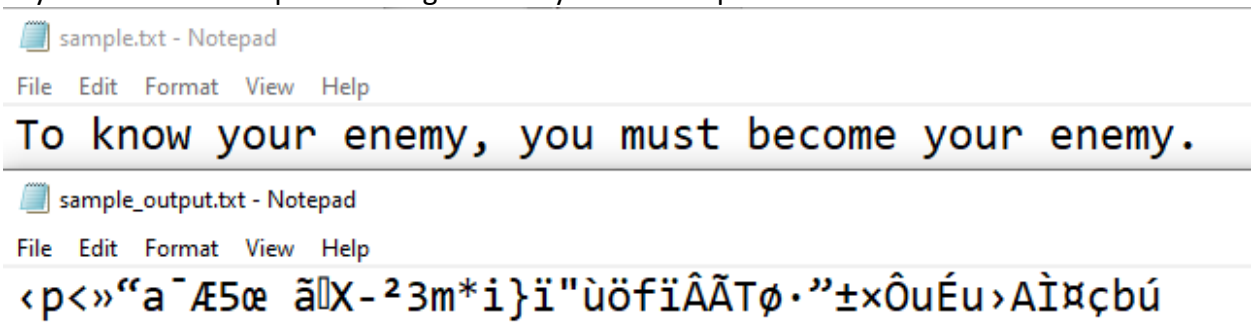
Here we can see two images side by side using CBC and the key 'bcit'.

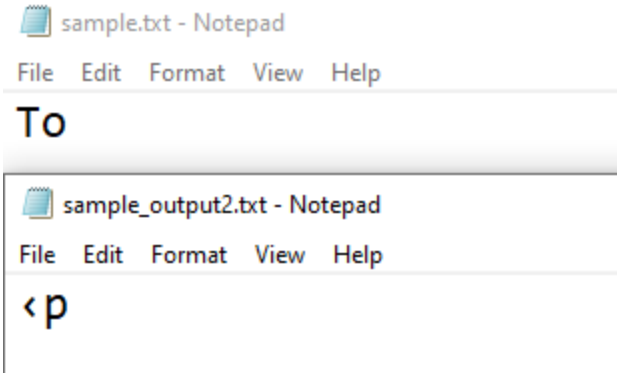
Unlike the previous image we looked at that was encrypted using ECB, it's extremely hard to find any patterns in the CBC encrypted image. Through the usage of 'pseudo random' IV, the confusion has been greatly improved upon. A weakness of CBC however is that the encryption itself is sequential however for the purposes of this, CBC has strong diffusion.

Confusion

Similar to ECB, when looking at the confusion side I'm looking at the relationship between the ciphertext and key and whether it's possible to get the key from the ciphertext.



Here we can see that it's impossible to see any relation between the ciphertext and the key which again was 'bcit'. There's no reasonable way to know what the ciphertext or key could be from observing the output. An issue I noticed when looking at the confusion was that the length of the sample text was the same as the length of the encrypted text, which can be an issue. Here is another example of what I mean.



With a smaller text needing to be encrypted the output would be the same length.

Avalanche Effect

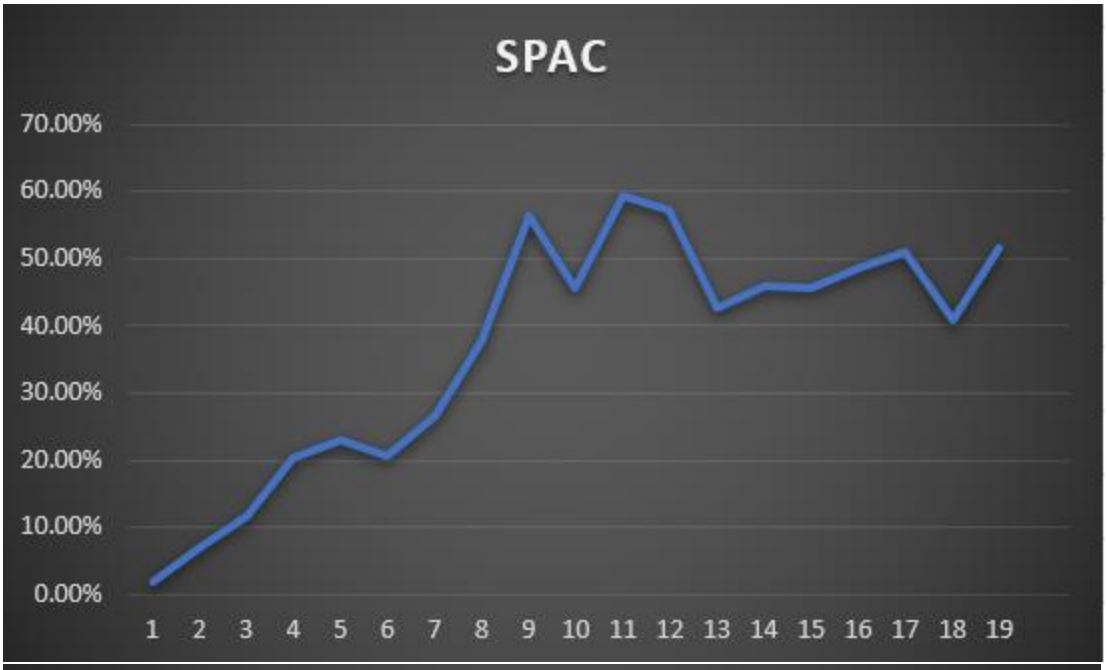
The following is a table of bits changed in the plaintext as well as changes in the key for SPAC.

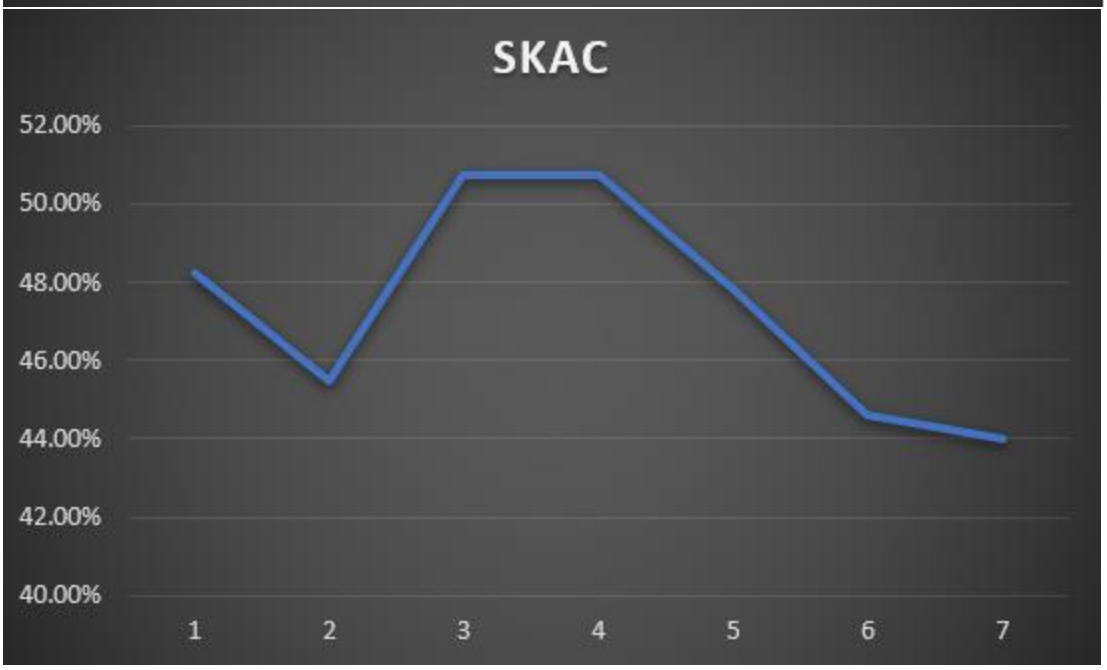
Round	Bits Changed in Cipher	Change Percentage
1	14	1.97%
2	48	6.84%
3	81	11.54%
4	135	20.27%
5	153	22.97%
6	139	20.59%
7	176	26.43%
8	253	37.48%
9	359	56.18%
10	303	45.50%
11	374	59.37%
12	351	57.35%
13	304	42.76%
14	346	45.77%
15	320	45.58%
16	327	48.44%
17	334	50.84%
18	293	40.69%
19	339	51.60%

As well as for SKAC.

Key	Bits Changed in Cipher	Change Percentage
duckduckgo	317	48.25%
duckduckgg	315	45.45%
duckdukkgg	315	50.72%
dvckdukkgg	315	50.72%
dvckdukpgg	297	47.83%
zuckdukpgg	297	44.59%
zvckdukpgg	313	44.02%

Converting both to line graphs produces the following:





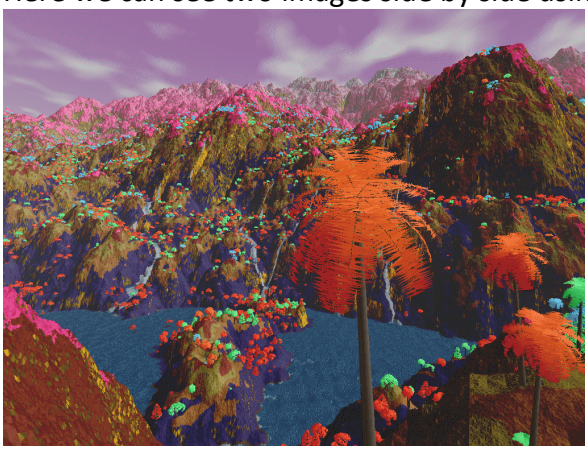
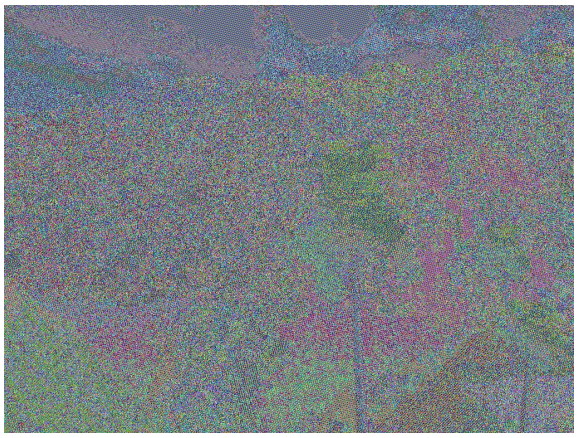
SKAC data was collected using the same plaintext but with a key with single bits changing throughout each interaction. Meanwhile, SPAC data is from modifying the plaintext one bit at a time using the same key throughout. Both graphs remains consistently between 40% to 60% outside of the initial bit changes for SPAC.

CTR (Counter)

CTR otherwise known as Counter mode essentially ensures that a sequence will not repeat for a very long time. The plaintext is substituted with different values every time, thus having a very good confusion property. Unlike the previous two modes covered, EBC and CBC, the encryption is done with a counter, key and nonce which is then XOR'd to produce the ciphertext.

Diffusion

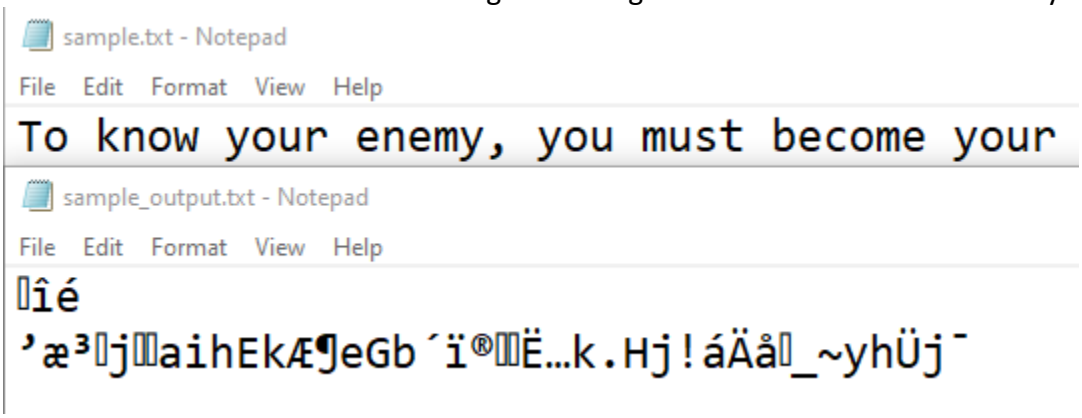
Here we can see two images side by side using CTR and the key 'bcit' and a counter number of 47.

From the image above, we can see that while it's better than when we encrypted the image using ECB, it's not as good as when we used CBC previously. This could be due to sacrificing speed for performance as running CTR has been faster than CBC due the encryption being parallelized (You're don't need to rely on other blocks so the encryption can be done in parallel).

Confusion

Similar to the methods above I'll be again looking at the relation between the key and ciphertext.



We can see that there's no discernable way to get the key from the given ciphertext. In addition to having a secret key, CTR also has a nonce which is shared between the sender and receiver. While the ciphertext is the same length as the plaintext similar to the methods above, the relation between having to use both a key and nonce adds to the confusion factor.

Avalanche Effect

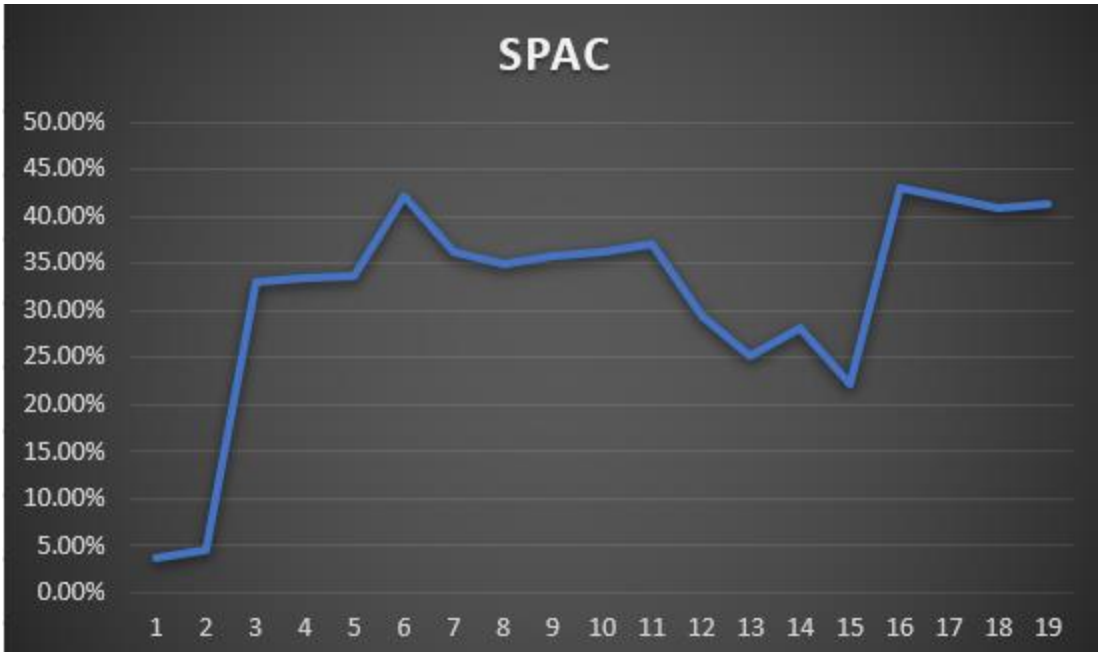
The following is a table of changes in bits for the plaintext (SPAC). I used the same counter value of 24.

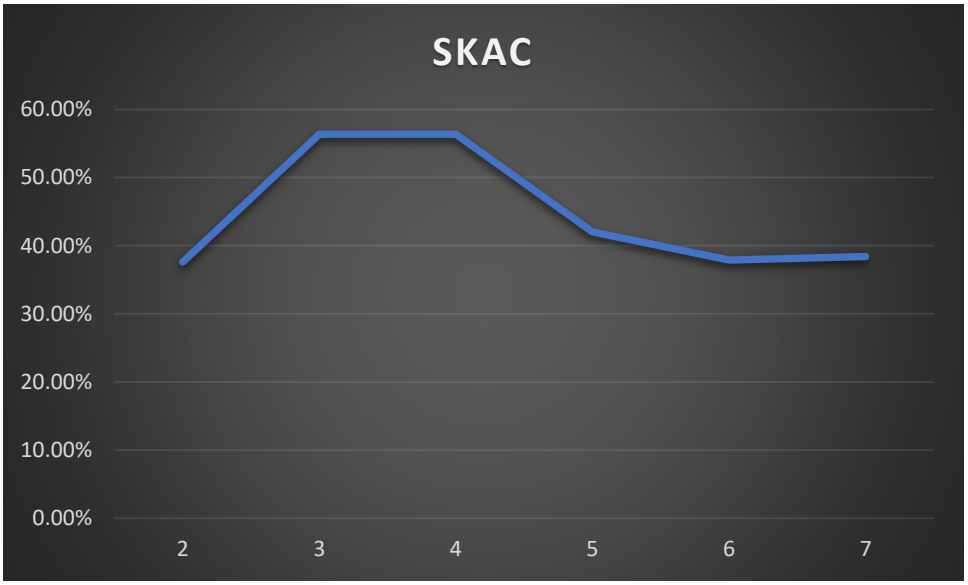
Round	Bits Changed in Cipher	Change Percentage
1	23	3.60%
2	29	4.48%
3	217	33.03%
4	222	33.33%
5	224	33.63%
6	280	42.04%
7	244	36.15%
8	232	34.83%
9	242	35.85%
10	241	36.19%
11	246	36.94%
12	198	29.33%
13	168	25.23%
14	187	28.08%
15	143	22.07%
16	283	43.07%
17	283	41.93%
18	276	40.89%
19	275	41.29%

As well as for changes in the bits for the key.

Key	Bits Changed in Cipher	Change Percentage
duckduckgo	317	48.25%
duckduckgg	315	45.45%
duckdukkgg	315	50.72%
dvckdukkgg	315	50.72%
dvckdukpgg	297	47.83%
zuckdukpgg	297	44.59%
zvckdukpgg	313	44.02%

For better visualization, here's line graphs for each one.



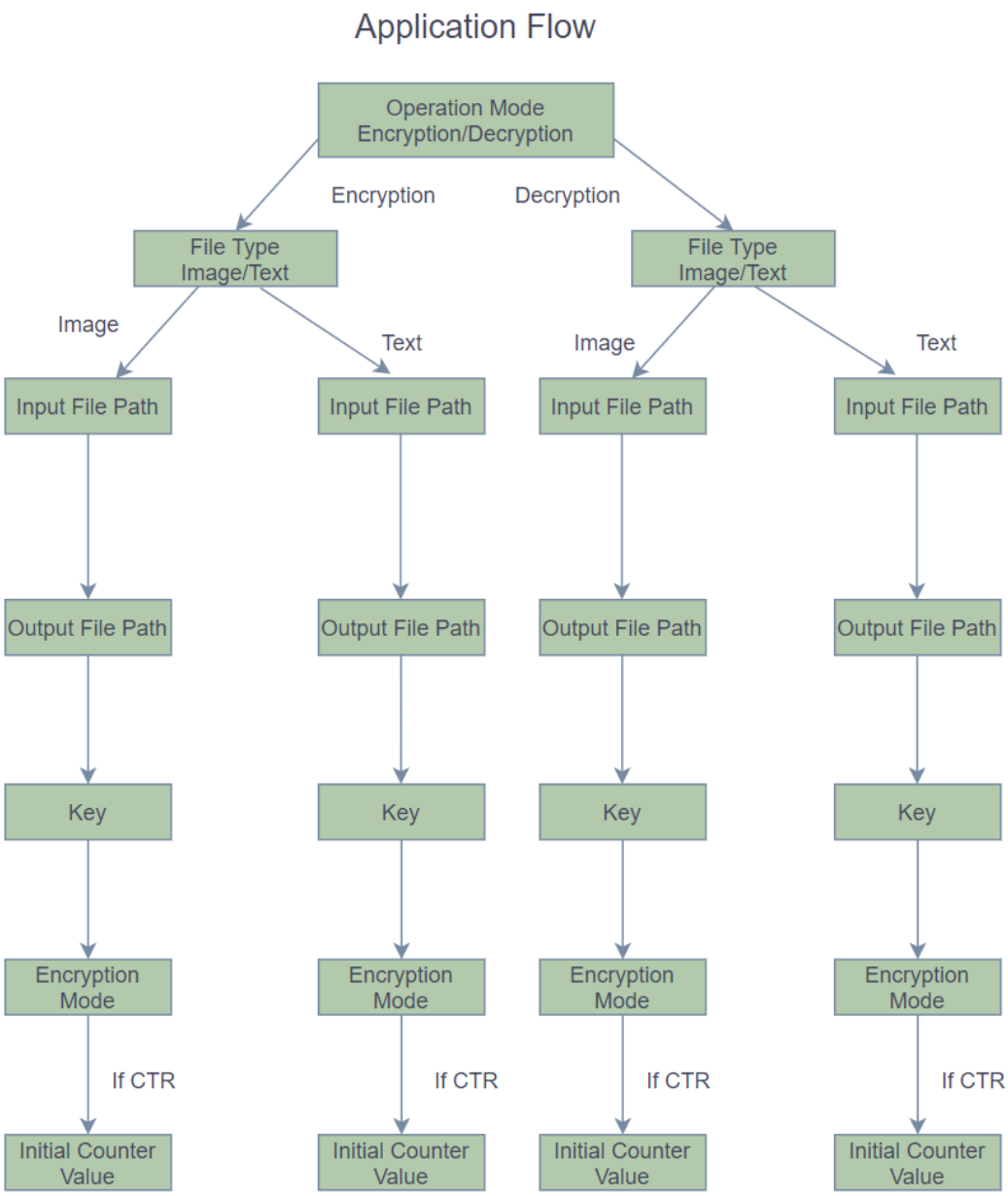


From the SPAC graph, we can see that the percentage in change is a fair bit lower from the initial data of up to 19-bit changes, but it starts to normalize at around 40%. On the other side, we can see the SKAC stay consistently between the 38% to 56% mark.

Design

Application Flow

As always, first let us look at the high-level application flow:



The application first prompts users for the operation mode, which can be encryption or decryption. And once user has picked the operation mode, then it asks for the file type of the input file; the reason of it is that the application performs different encryption and decryption processes based on the type of the file in order to avoid manipulating the original file header.

After confirming the file type, the application will then prompt user for entering the input and output file paths. The user will have to choose one of the following three encryption modes (Detailed design of each encryption mode will be explained in the next section):

1. ECB mode

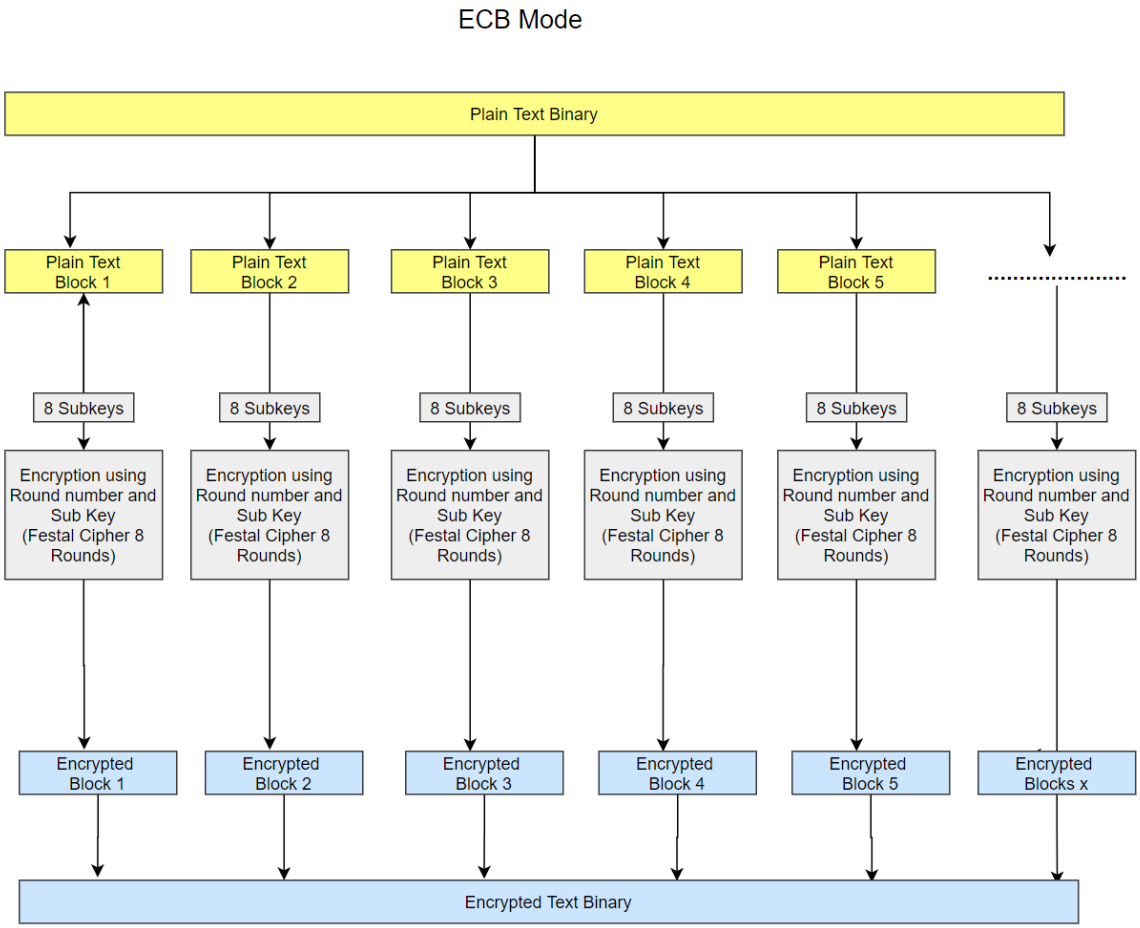
2. CBC mode
3. CTR mode (User also needs to enter initial counter value)

Once the encryption mode is chosen, the application then encrypts or decrypts the inputted file using the chosen encryption mode and saves the encrypted or decrypted data as the output file.

Encryption Modes

ECB

Let us take a look at the ECB mode encryption design:

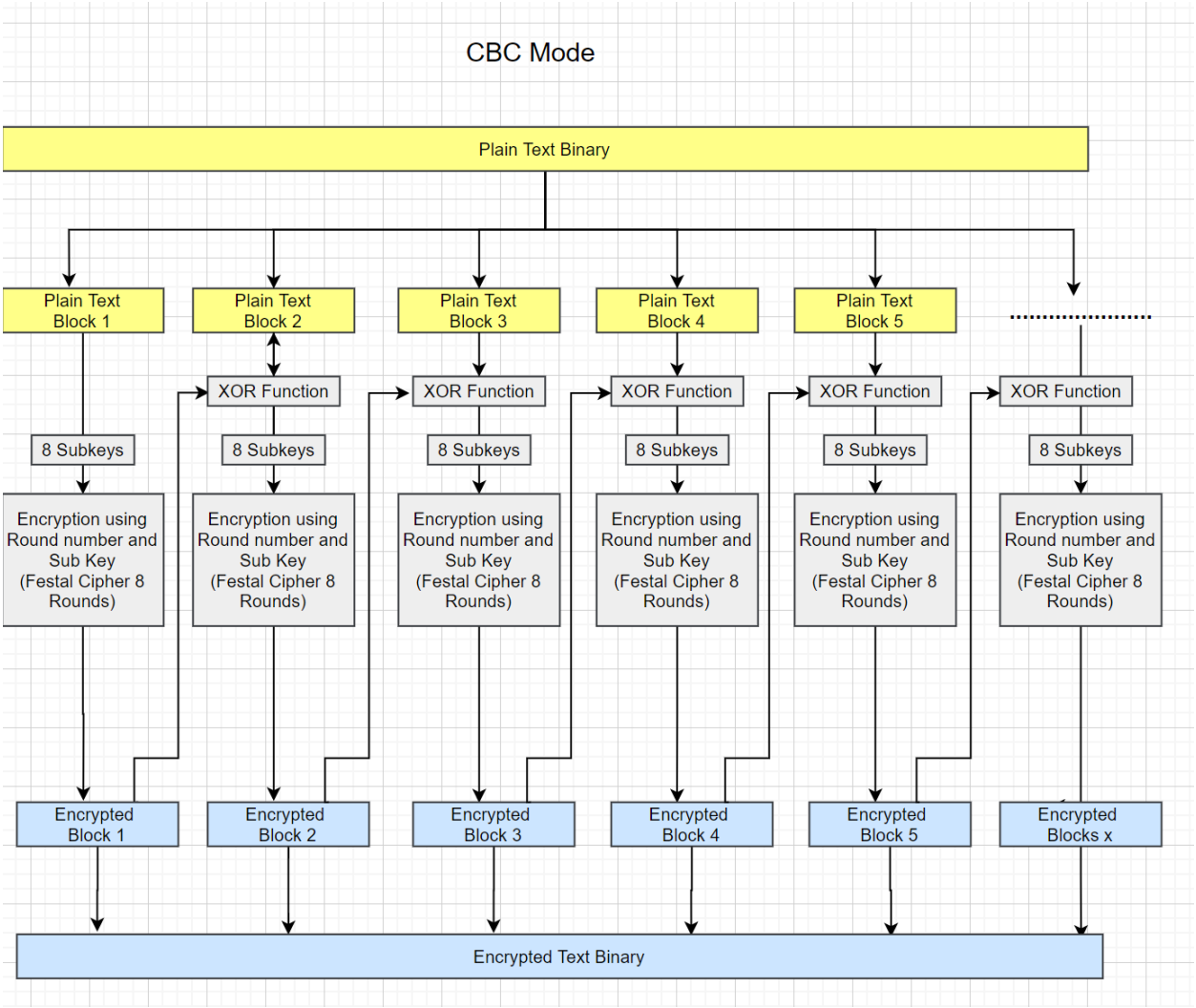


ECB is the most basic block cipher encryption mode. When using ECB encryption mode, the application will simply divide the original plain text binary into small binary blocks, and each binary block will get encrypted using the same encryption process. At the end, these small encrypted binary blocks will be put back together to form the encrypted cipher message or image.

Problem with this encryption mode is that each of the binary blocks will go through the exact same encryption process, thus the confusion and diffusion of the encryption is hugely impacted by the block size of the binary blocks.

CBC

Next, we'll look at the CBC mode encryption design:

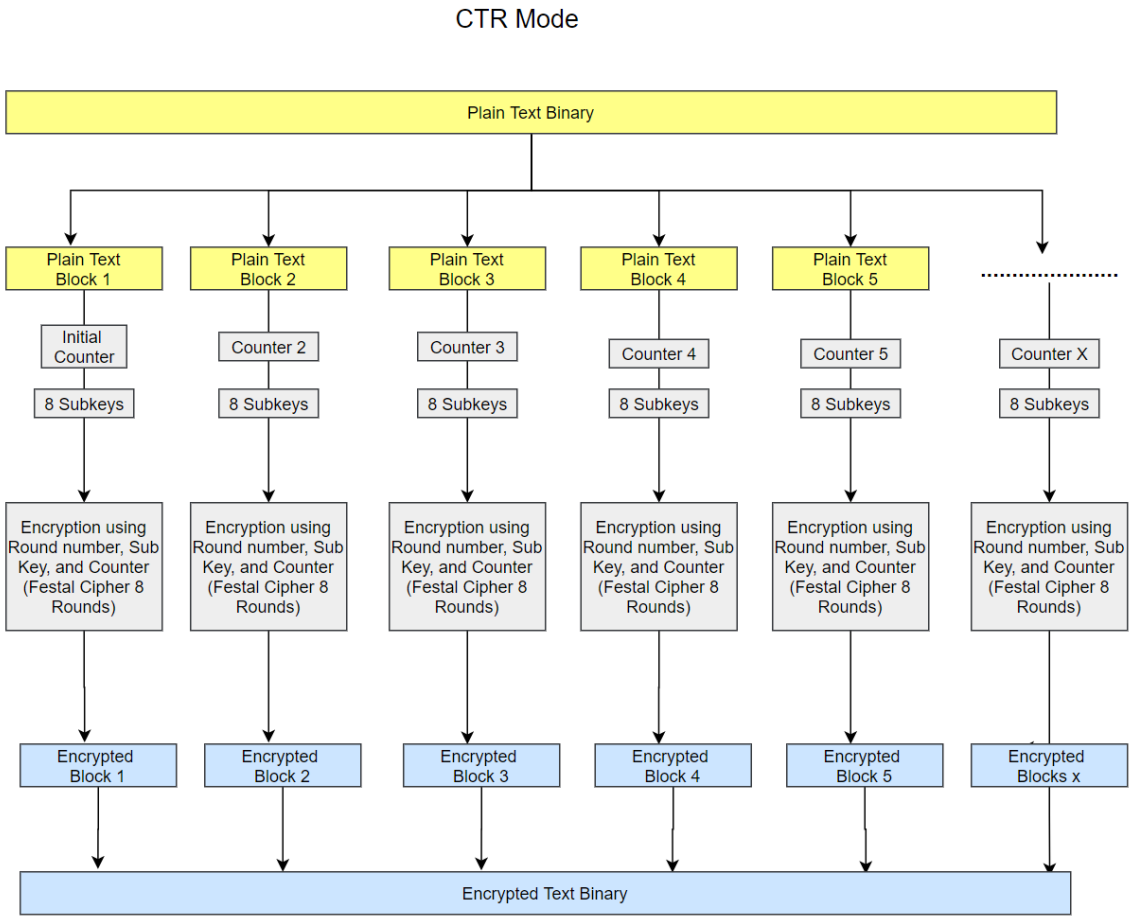


CBC mode stands for Cipher Block Chaining mode. From its name, we probably can tell it functions by chaining all binary blocks together.

CBC mode works very similar to the ECB mode, the only major difference is before each plaintext binary block gets encrypted, it will first get an XOR comparison against the encrypted binaries of the previous block. As the result, each binary block will be one of the encryption parameters of the next binary block, so almost all binary blocks will get encrypted uniquely. And therefor, the CBC mode will have a much better confusion than the ECB mode.

CTR

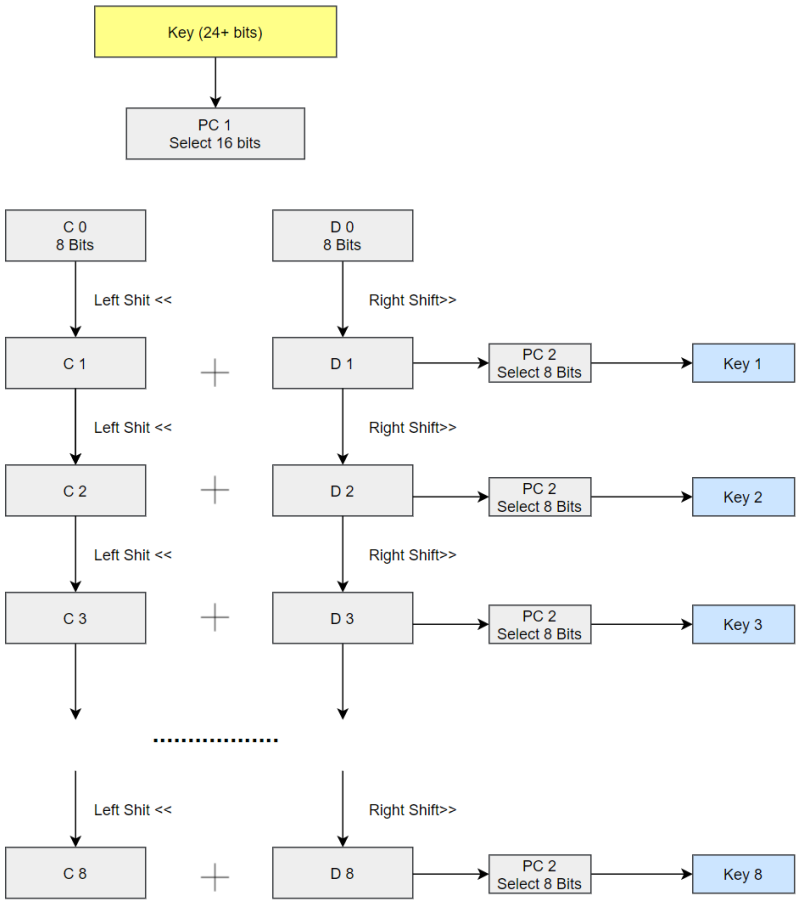
Here is the design for CTR mode:



CTR mode stands for Counter mode. It is also very similar to the ECB mode. The major difference between the CTR mode and the ECB mode is that instead of only using the subkeys or subkeys and round numbers to encrypt each binary block, the CTR mode also uses the counters as an extra encryption input. Usually, user will have to pick an initial counter value, and the system will modify the counter value for each different binary blocks. Therefore, the confusion and diffusion will get hugely improved when comparing to the ECB mode.

Subkeys Generation

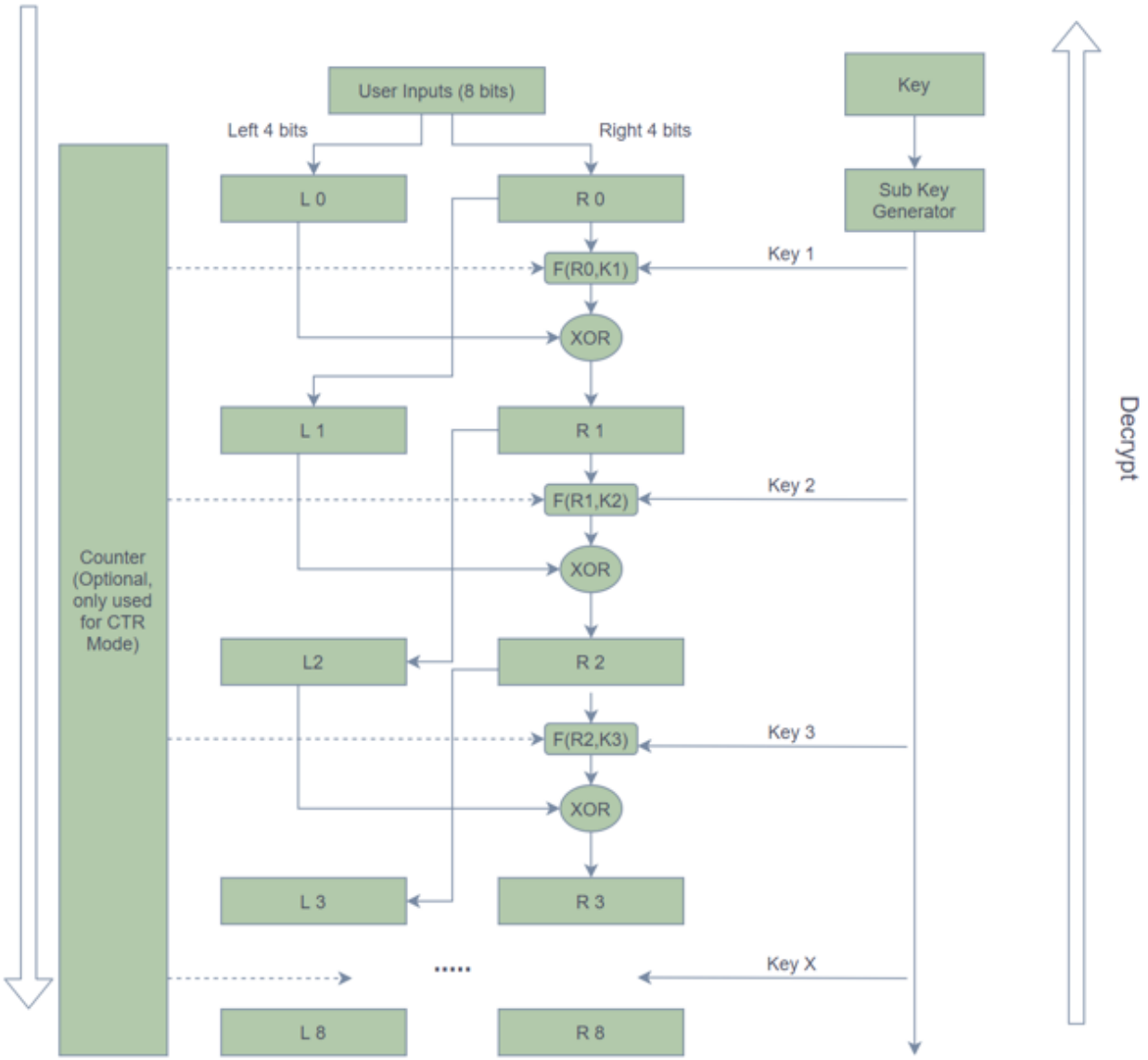
All of the encryption modes above uses subkeys as one of the encryption inputs. Here is the design:



As we can see in the above graph, a key which is 24 bits or longer will be passed to the sub key generation function. (Only for this application, standard DES often uses 64 bits long key)
And the function will first pick 16 bits from these 24 bits based on the filter called PC1 (Permuted Choice 1, which defines which bits will get processed).
Next, the function divides the 24 bits into C and D sections, each section will have 8 bits.
Now to generate a new subkey, the function will left shift the binary inside the C section (So 00010000 will become 00100000) and right shift the binary inside the D section (So 00010000 will become 00001000). Then it concatenates the C and D sections and uses PC2 (Permuted Choice 2) to further filter the binary down to 8 bits. The 8 bits long binary is the final subkey.
And to generate 8 subkeys, the function will repeat the same processes 8 times.

Feistel Encryption Flow

Next, let's look at the Feistel encryption flow design:



It has a scrambling function of:

$$fi(x,k) = [(2i * k + c)^x] \; mod \; 15$$

Where i is the number of the round; and K is the encryption Key (Any integer).
Where c is the counter, which will only be a non-zero number when using CTR mode.

Like any other Feistel encryption algorithms, the application first splits the plain text binary block into two parts the left part - L and the right part – R. Next, the application will add the value in a current round of R to the next round of L. It will also pass the value into the current round of R to the scrambling function. The scrambling function encrypts it based on the number of the current round, generated sub key and the counter number (if applicable, only for CTR mode). The application then uses the result from the scrambling function to perform a XOR comparison against the value in the current round of L. The output of the XOR comparison will then be added to the next round of R.

To summarize the Feistel encryption process of each round, we can use the following two equations:

$L_i = R_{i-1}$

$R_i = L_i \oplus F(R_{i-1}, K_i)$

Currently, the application is configured to perform 8 rounds of the Feistel encryption process, but which can be changed anytime.

The decryption process is also very straightforward, the application will simply reverse the above Feistel encryption processes to decrypt the encrypted files.

Testing

My setup for testing is a Windows 10 computer, where I'll be IDLE to run the script.

Note that for Test 1 and Test 2, the videos in the 'Videos' folder are significantly longer due to problems with deleting the waiting portion of the encryption and decryption of images. The issue was resolved for Test 3 and onwards.

Test #	Tools Used	Expectation	Actuality	Result
1	IDLE	I should be able to encrypt the Plaintext and Image file using a key that I can choose with ECB	I was able to encrypt the files using a key I chose	PASS
2	IDLE	I should be able to decrypt the encrypted Plaintext and Image from Test #1 using the same key with ECB	I was able to decrypt the files from Test #1 using the same key	PASS
3	IDLE	I shouldn't be able to decrypt the encrypted Plaintext and Image from Test #1 using a different key with ECB	I wasn't able to decrypt the files from Test #1 using the key that wasn't used for the encryption originally	PASS
4	IDLE	I should be able to encrypt the Plaintext and Image file using a key that I can choose with CBC	I was able to encrypt the files using a key I chose	PASS
5	IDLE	I should be able to decrypt the encrypted Plaintext and Image from Test #4 using the same key with CBC	I was able to decrypt the files from Test #4 using the same key	PASS
6	IDLE	I shouldn't be able to decrypt the encrypted Plaintext and Image from Test #4 using a different key with CBC	I wasn't able to decrypt the files from Test #4 using the key that wasn't used for the encryption originally	PASS
7	IDLE	I should be able to encrypt the Plaintext and Image file using a key that I can choose with CTR	I was able to encrypt the files using a key I chose	PASS
8	IDLE	I should be able to decrypt the encrypted Plaintext and Image from Test #5 using the same key with CTR	I was able to decrypt the files from Test #5 using the same key	PASS
9	IDLE	I shouldn't be able to decrypt the encrypted Plaintext and Image from Test #5 using a different key with CTR	I wasn't able to decrypt the files from Test #5 using the key that wasn't used for the encryption originally	PASS

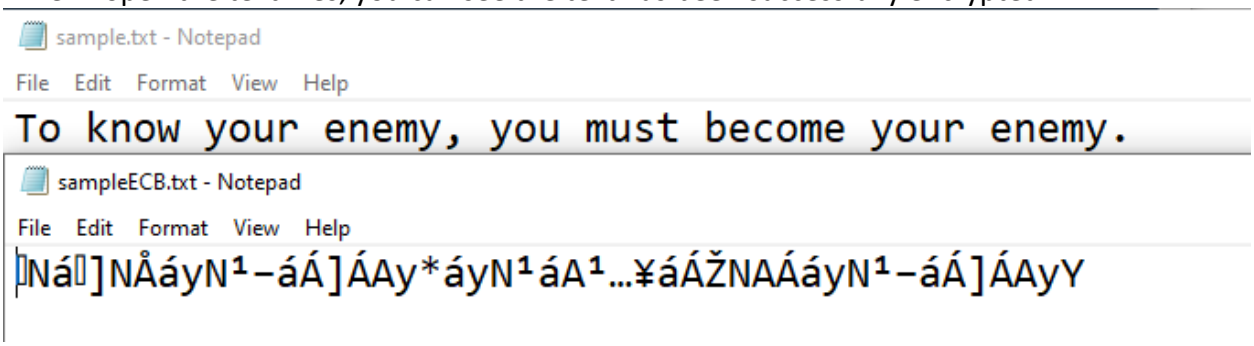
Test 1

For this test, I should be able to encrypt a text and image file using a key that I can choose with ECB.
For all of the text portion, I'll be using a text file called sample.txt with the following line in it, "To know your enemy, you must become your enemy" for all the tests moving forward.

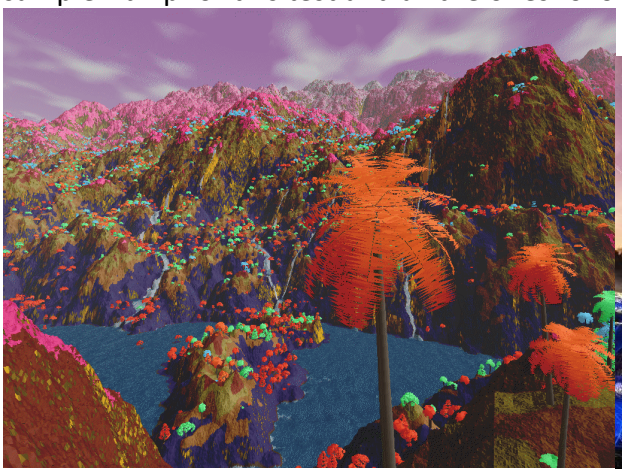
When running the script from idle, I'll be choosing "1" for encryption mode, "2" for text file, the key will be "canada2020" and the encryption mode will be "1" for ECB.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sample.txt
Please enter the output file name/path: sampleECB.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
------------------------------------------------------------------------------------------------
 Encrypting, please wait
------------------------------------------------------------------------------------------------
376
Finishsed encryption, please check the folder
```

When I open the text files, you can see the text has been successfully encrypted.



For the image encryption portion, I'll be using the following two BMP images labeled sample1.BMP and sample2.bmp for this test and all the ones following it.

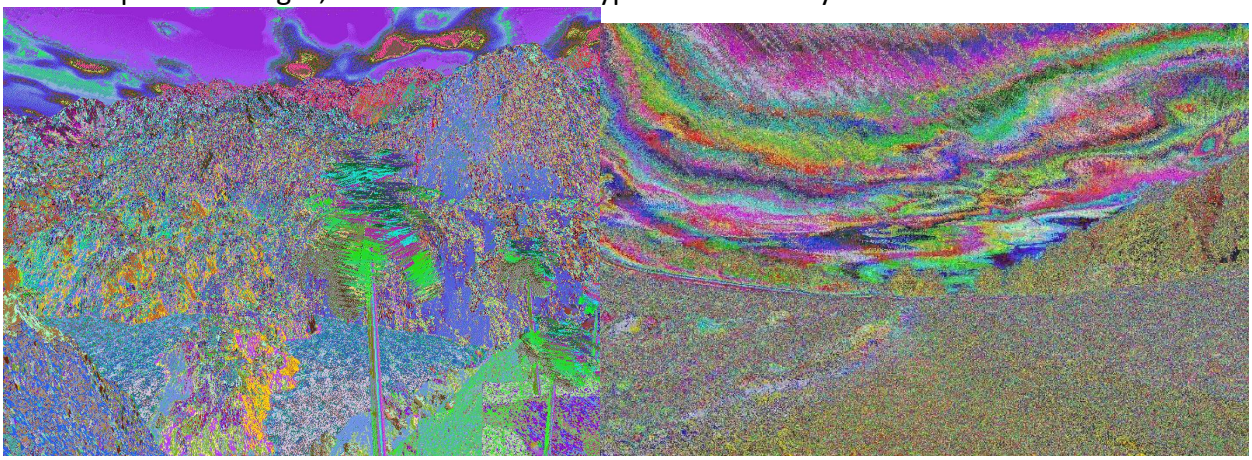


Similar to the text encryption above, the only difference is that I've chosen "1" for Image file.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample1.BMP
Please enter the output file name/path: sample1ECB.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
-----------------------------------------------------------------------------------------------
 Encrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finishsed encryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample2.bmp
Please enter the output file name/path: sample2ECB.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
-----------------------------------------------------------------------------------------------
 Encrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finishsed encryption, please check the folder
```

When I open the images, both look to be encrypted successfully.



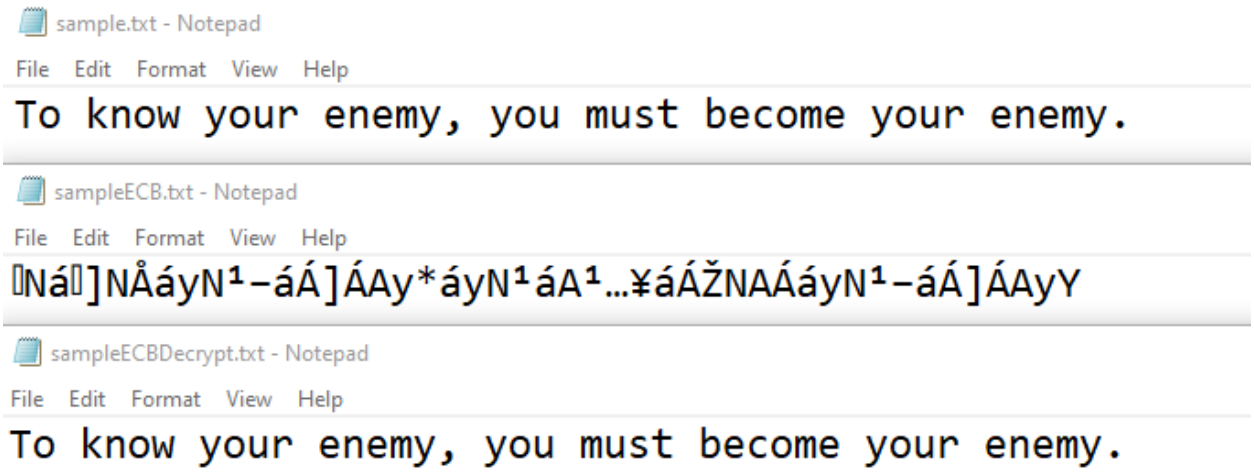
Test 2

Here I'll be testing the decryption function for the encrypted files from Test #1 using the same key as was used during the encryption.

The following were the steps taken to decrypt the text.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sampleECB.txt
Please enter the output file name/path: sampleECBDecrypt.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
-----------------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------------
Finished decryption, please check the folder
```

As shown below, the text file has been successfully decrypted to the original text file.

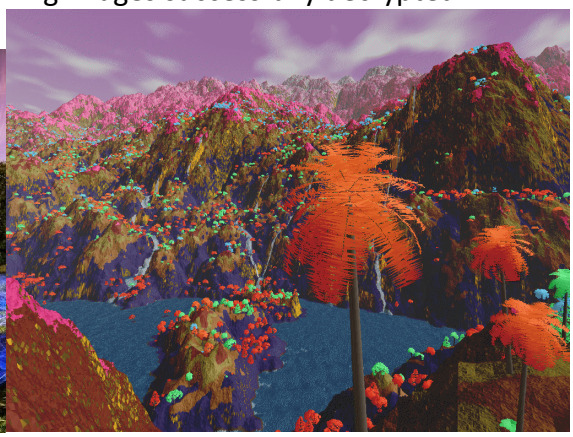


Now I'll be doing the same process for the two images.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample1ECB.BMP
Please enter the output file name/path: ECBDecrypt.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample2ECB.BMP
Please enter the output file name/path: ECBDecrypt2.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

When we open the images, we get the following images successfully decrypted.



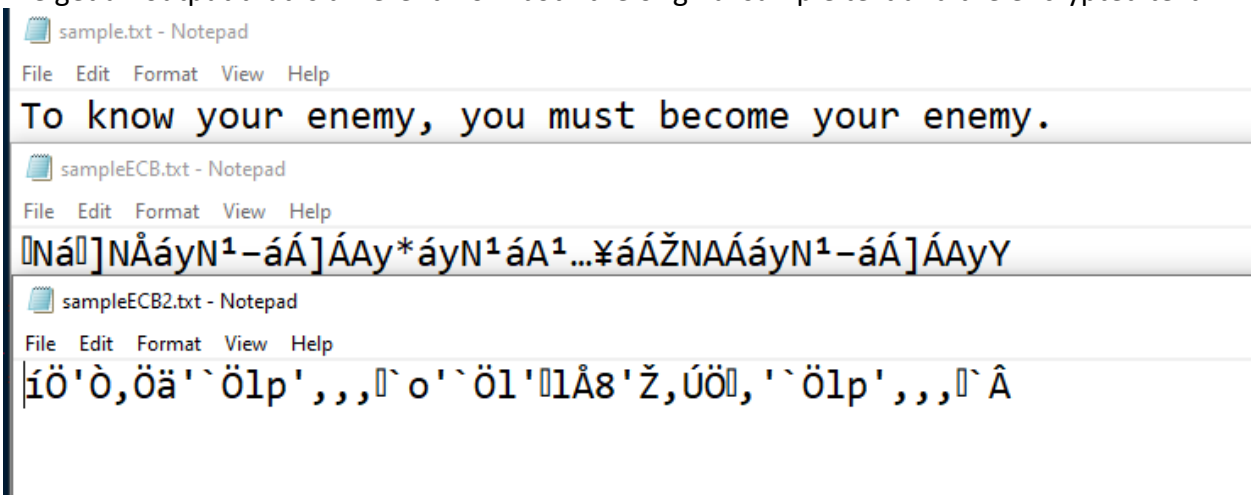
Test 3

Here I'll test if using a key that isn't the same as the one I used to encrypt would decrypt the files.

I used 'british123' as the key which is different from the original key, 'canada2020'.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sampleECB.txt
Please enter the output file name/path: sampleECB2.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): british123
Generate Subkey:32
Generate Subkey:16
Generate Subkey:1
Generate Subkey:160
Generate Subkey:74
Generate Subkey:132
Generate Subkey:8
Generate Subkey:87
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Finished decryption, please check the folder
```

We get an output that is different from both the original sample text and the encrypted text.

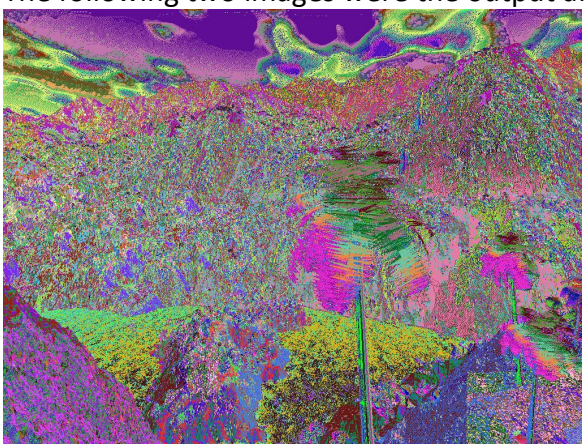


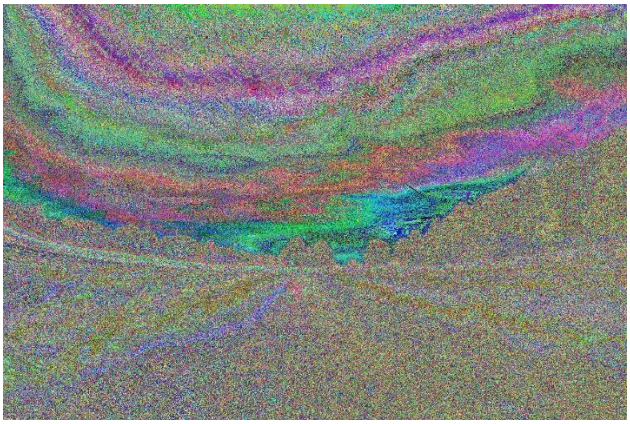
Now for the same process but for the images. For the first image, I used 'halifax992' as the key and 'alberta9923' for the second image, both image original key being 'canada2020'.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample1ECB.bmp
Please enter the output file name/path: sample1ECB1.bmp
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): halifax992
Generate Subkey:238
Generate Subkey:79
Generate Subkey:181
Generate Subkey:251
Generate Subkey:127
Generate Subkey:36
Generate Subkey:211
Generate Subkey:251
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample2ECB.BMP
Please enter the output file name/path: sample2ECB2.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): alberta9923
Generate Subkey:16
Generate Subkey:128
Generate Subkey:74
Generate Subkey:132
Generate Subkey:136
Generate Subkey:29
Generate Subkey:36
Generate Subkey:16
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 1
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

The following two images were the output after trying to encrypt.





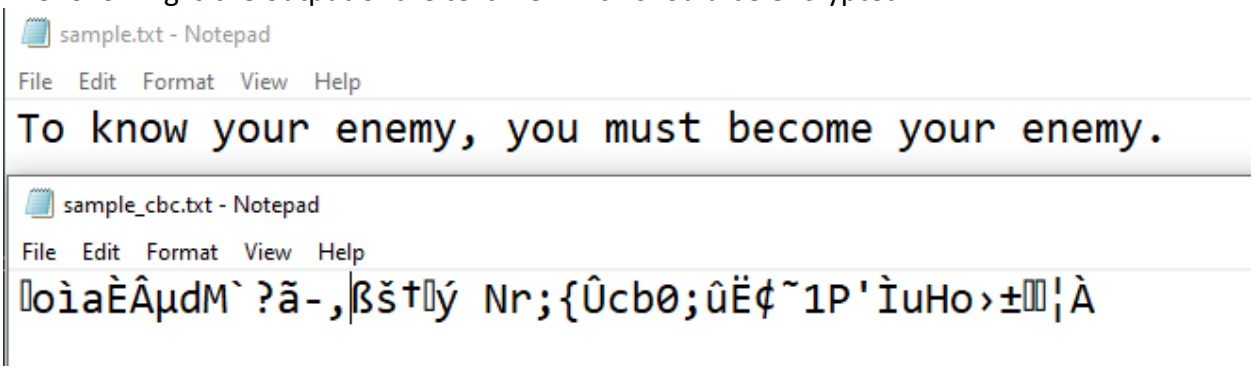
Thus, showing that without the correct key, you cannot decrypt the image.

Test 4

Here I'll be repeating what I did for Test #1 but using CBC mode, starting off with the text file. I'll be using the same key, 'canada2020'.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sample.txt
Please enter the output file name/path: sample_cbc.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 2
-----------------------------------------------------------------------------------------------
 Encrypting, please wait
-----------------------------------------------------------------------------------------------
376
Finishsed encryption, please check the folder
```

The following is the output of the text file which should be encrypted.

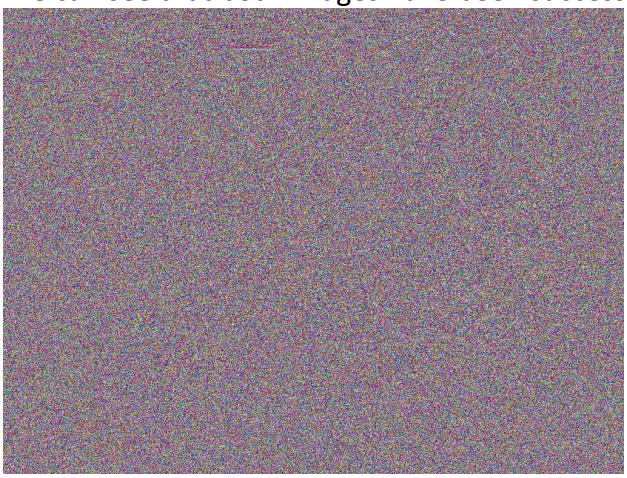


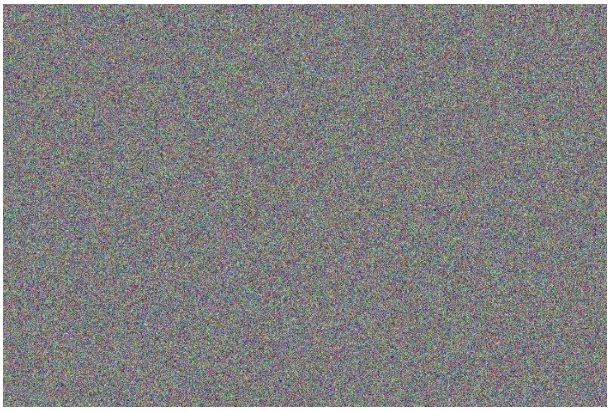
Which we can see is encrypted successfully using CBC. Now we'll do the same process but for the two images.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample1.BMP
Please enter the output file name/path: sample1CBC.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 2
-----------------------------------------------------------------------------------------------
 Encrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finishsed encryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample2.bmp
Please enter the output file name/path: sample2CBC.bmp
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 2
-----------------------------------------------------------------------------------------------
 Encrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finishsed encryption, please check the folder
```

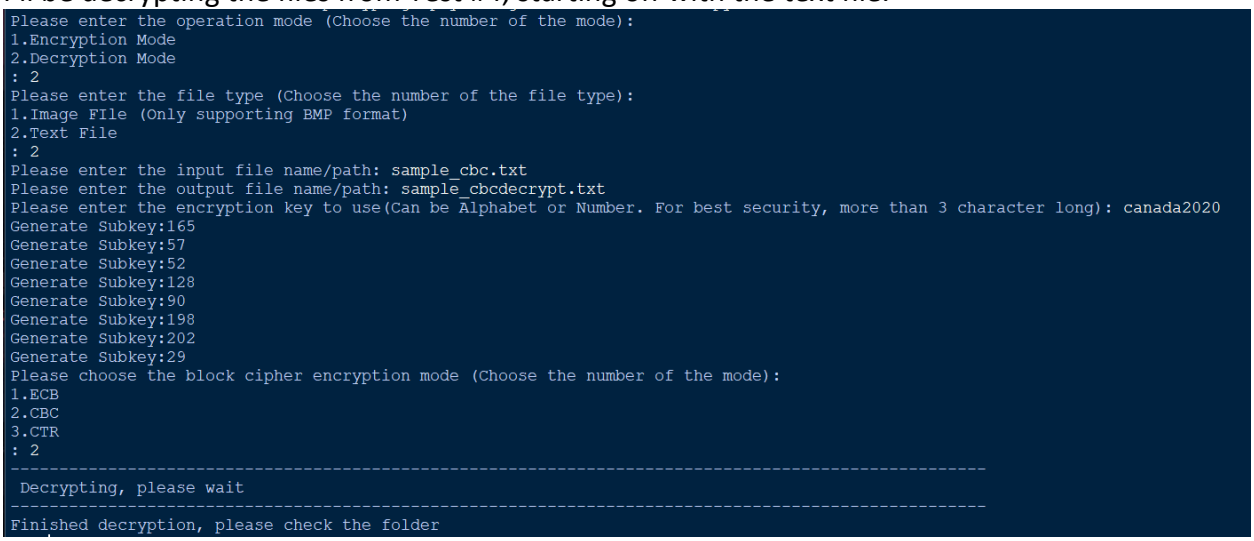
We can see that both images have been successfully encrypted below.



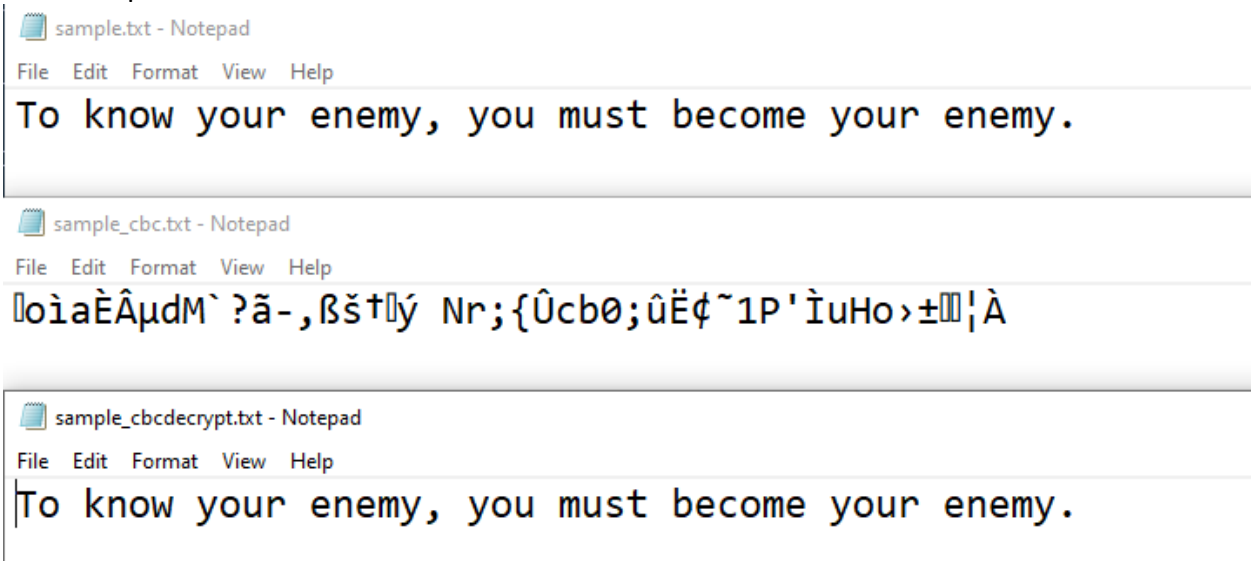


Test 5

I'll be decrypting the files from Test #4, starting off with the text file.



The decrypted image is as follows, showing that the text has been successfully decrypted to the original text from sample.txt.

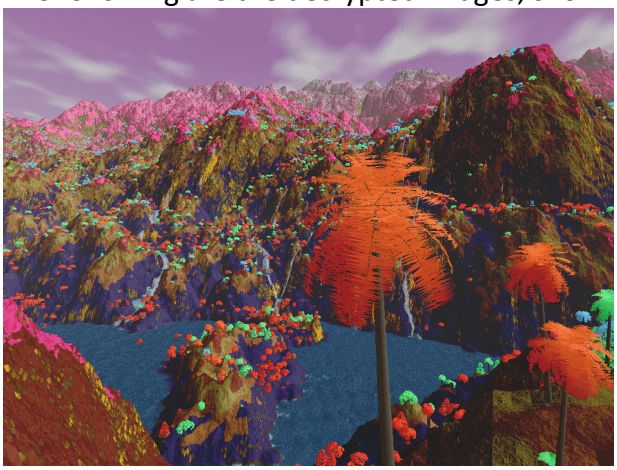


I'll be doing this again for both the images now.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample1CBC.BMP
Please enter the output file name/path: sample1CBCDECRYPT.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 2
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample2CBC.BMP
Please enter the output file name/path: sample2CBCDECRYPT.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 2
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

The following are the decrypted images, showing that the images were successfully decrypted.



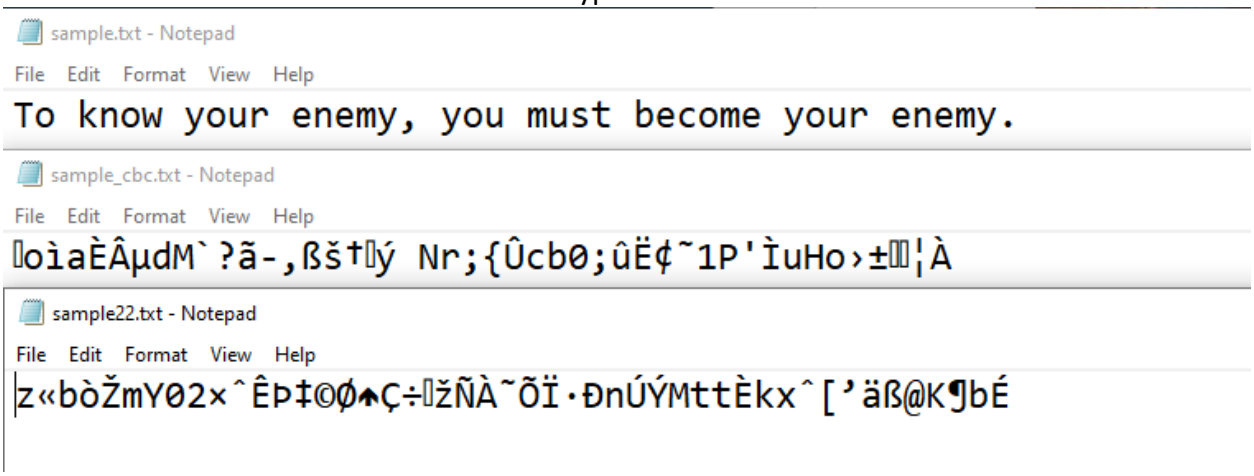
Test 6

Here I'll show whether using a different key would be able to decrypt the encrypted files, starting off with the text.

Here I'll be using a key of 'jinnathom9'.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sample_cbc.txt
Please enter the output file name/path: sample22.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): jinnathom9
Generate Subkey:199
Generate Subkey:106
Generate Subkey:215
Generate Subkey:41
Generate Subkey:53
Generate Subkey:99
Generate Subkey:176
Generate Subkey:74
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 2
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Finished decryption, please check the folder
```

Below we can see that the text was not decrypted.

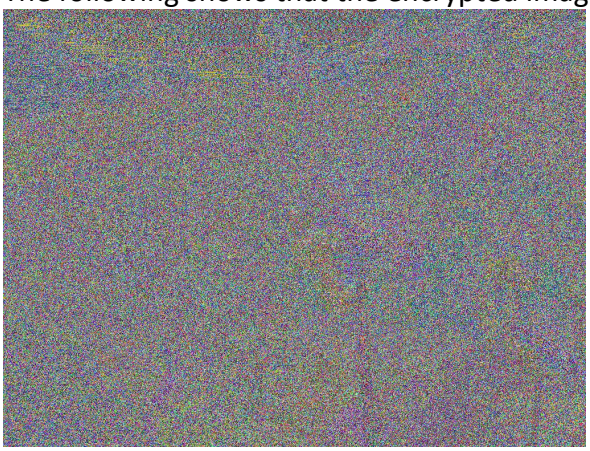


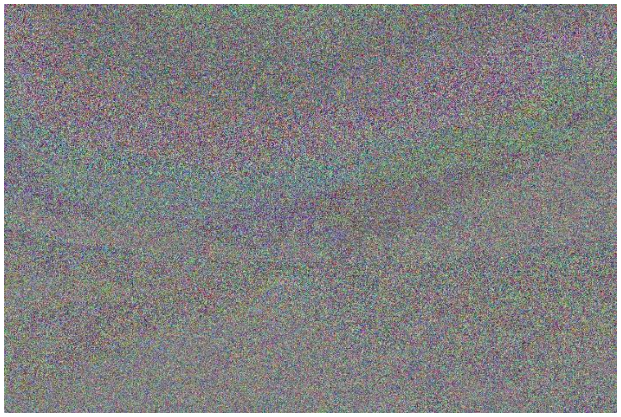
Next I'll be doing it for both images with the key "onthehill92" and "britishcolumbia".

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample1CBC.BMP
Please enter the output file name/path: sample11.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): onthehill92
Generate Subkey:66
Generate Subkey:66
Generate Subkey:1
Generate Subkey:99
Generate Subkey:33
Generate Subkey:32
Generate Subkey:1
Generate Subkey:98
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 2
-----------------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample2CBC.bmp
Please enter the output file name/path: sample222222.bmp
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): britishcolumbia
Generate Subkey:168
Generate Subkey:12
Generate Subkey:132
Generate Subkey:218
Generate Subkey:94
Generate Subkey:4
Generate Subkey:17
Generate Subkey:177
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 2
-----------------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

The following shows that the encrypted images were not able to get decrypted successfully.



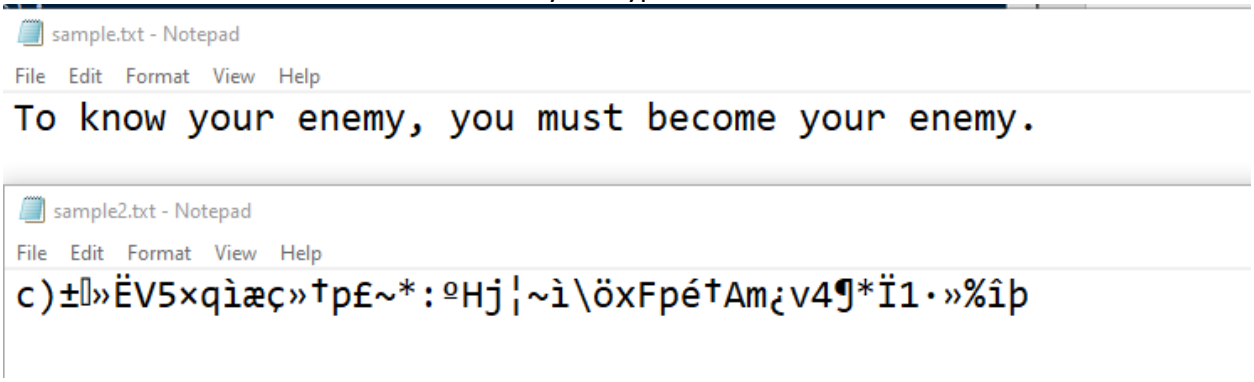


Test 7

Here I'll be encrypting using CTR. I'll be using the same key "canada2020" and a counter value of 27.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sample.txt
Please enter the output file name/path: sample2.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number (Any integer number):27
-----------------------------------------------------------------------------------------------------
 Encrypting, please wait
-----------------------------------------------------------------------------------------------------
376
Finishsed encryption, please check the folder
```

Below we can see the text was successfully encrypted.

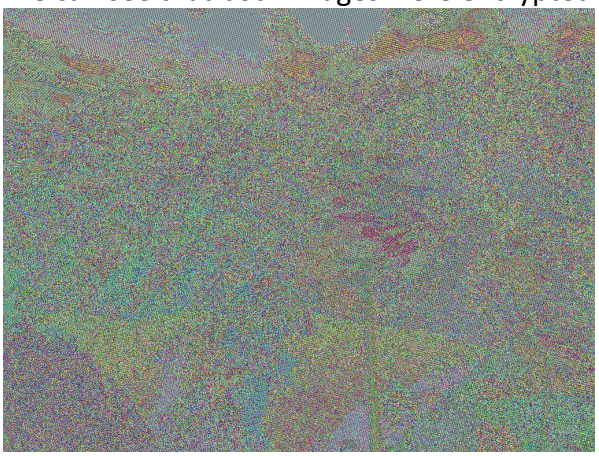


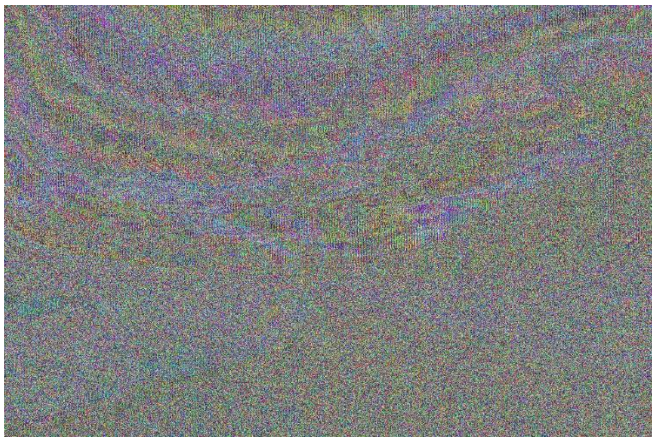
Now the same procedure but for the images.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample1.BMP
Please enter the output file name/path: sample11.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number (Any integer number):27
-----------------------------------------------------------------------------------------------------
 Encrypting, please wait
-----------------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finishsed encryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 1
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample2.bmp
Please enter the output file name/path: sample22.bmp
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number (Any integer number):27
-----------------------------------------------------------------------------------------------
 Encrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finishsed encryption, please check the folder
```

We can see that both images were encrypted successfully below.



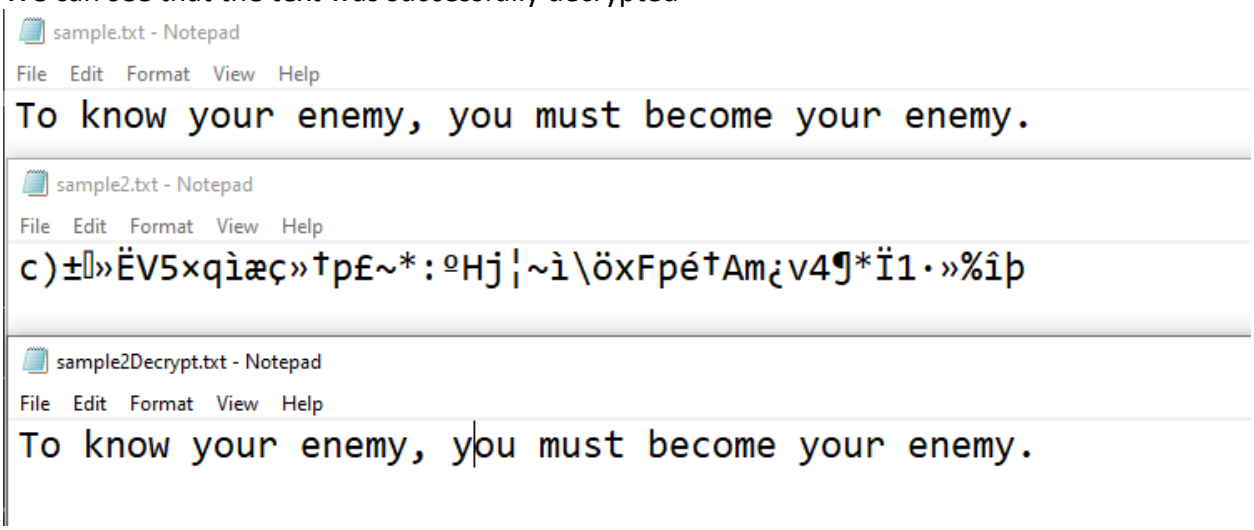


Test 8

Here we'll decrypt the files that we encrypted in Test #7 using the key 'canada2020' and counter value of 27.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sample2.txt
Please enter the output file name/path: sample2Decrypt.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number:27
--------------------------------------------------------------------------------------------------
 Decrypting, please wait
--------------------------------------------------------------------------------------------------
Finished decryption, please check the folder
```

We can see that the text was successfully decrypted

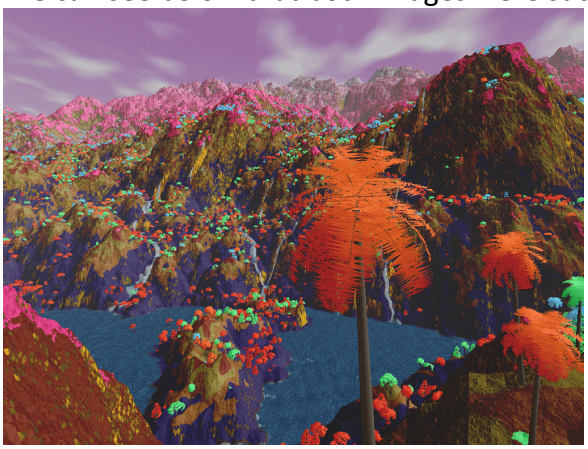


I'll repeat the process for the images.

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample11.BMP
Please enter the output file name/path: sample11Decrypt.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number:27
--------------------------------------------------------------------------------------------------
 Decrypting, please wait
--------------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample22.bmp
Please enter the output file name/path: sample22Decrypt.bmp
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number:27
--------------------------------------------------------------------------------------------------
 Decrypting, please wait
--------------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

We can see below that both images were successfully decrypted.



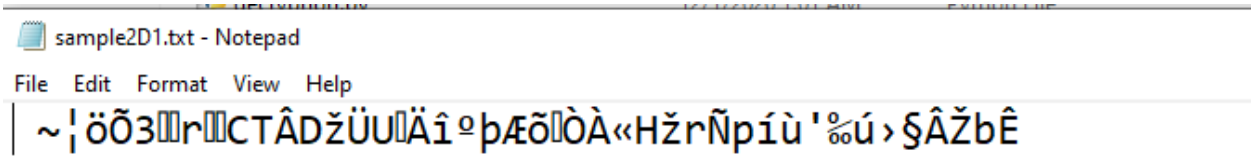
Test 9

Here I'll try to decrypt the encrypted files using a key that is different from the one used to encrypt it initially. I'll do two tests, one with the same key but different counter value and another with different key but same counter value.

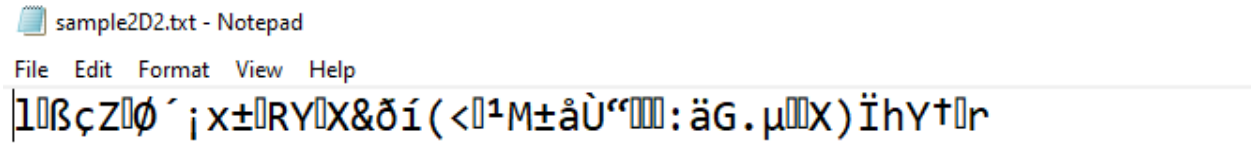
```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sample2.txt
Please enter the output file name/path: sample2D1.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): feistalcipher3
Generate Subkey:189
Generate Subkey:188
Generate Subkey:223
Generate Subkey:188
Generate Subkey:222
Generate Subkey:223
Generate Subkey:188
Generate Subkey:223
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number:27
------------------------------------------------------------------------------------------------
 Decrypting, please wait
------------------------------------------------------------------------------------------------
Finished decryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 2
Please enter the input file name/path: sample2.txt
Please enter the output file name/path: sample2D2.txt
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number:56
------------------------------------------------------------------------------------------------
 Decrypting, please wait
------------------------------------------------------------------------------------------------
Finished decryption, please check the folder
```

Below is the result from using the key 'feistalcipher' and the same counter value as the one when encrypting, 27.



Below is the result from using the same key 'canada2020' but a counter value of 56.



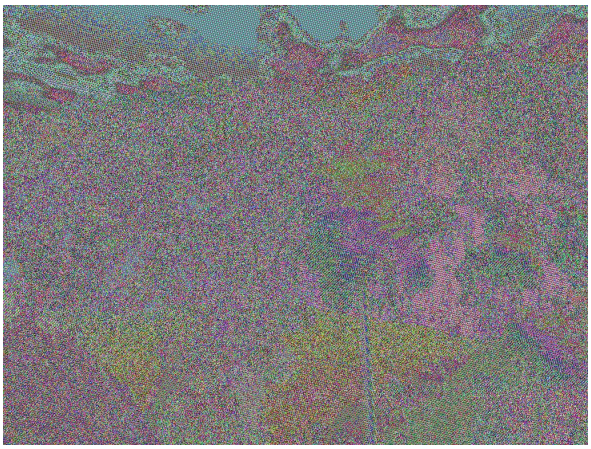
We can see in both cases, the decryption failed.

Now I'll do the same process but for the images. For the first image I'll use the same key but a different counter value. For the second image, I'll do the opposite with a different key but the same counter value.

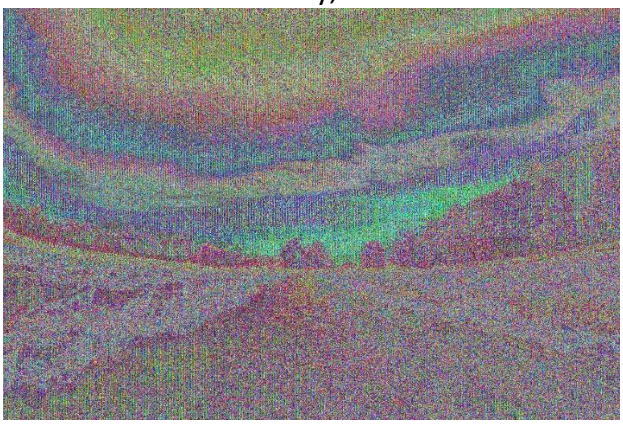
```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample11.BMP
Please enter the output file name/path: sample11DECRYPT.BMP
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): britishhills9
Generate Subkey:106
Generate Subkey:21
Generate Subkey:98
Generate Subkey:211
Generate Subkey:106
Generate Subkey:133
Generate Subkey:107
Generate Subkey:181
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number:27
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

```
Please enter the operation mode (Choose the number of the mode):
1.Encryption Mode
2.Decryption Mode
: 2
Please enter the file type (Choose the number of the file type):
1.Image FIle (Only supporting BMP format)
2.Text File
: 1
Please enter the input file name/path: sample22.bmp
Please enter the output file name/path: sample22DECRYPT.bmp
Please enter the encryption key to use(Can be Alphabet or Number. For best security, more than 3 character long): canada2020
Generate Subkey:165
Generate Subkey:57
Generate Subkey:52
Generate Subkey:128
Generate Subkey:90
Generate Subkey:198
Generate Subkey:202
Generate Subkey:29
Please choose the block cipher encryption mode (Choose the number of the mode):
1.ECB
2.CBC
3.CTR
: 3
Please enter the inital counter number:34
-----------------------------------------------------------------------------------------------
 Decrypting, please wait
-----------------------------------------------------------------------------------------------
Skipping header: 54bytes long
Finished decryption, please check the folder
```

Below I used a different key, "britishhills9" but the same counter value as when encrypted the image originally "27".



Here I used the same key, “canada2020” but a different counter value, “34”



We can see in both cases; the image was not successfully decrypted.