Experiment Report

Firstly, I defined MyAffCrypto to use AlphabeticStrings, then I wrote a plaintext in a text file in my Computer and I uploaded this file on cloud.sagemath. After that, I needed to read plaintext so I assigned it to read_data and encode utf-8. In order to have capital letters and text without space for encryption, I encoded read_data and assigned to P_Text. Then, I generated two random keys and used them in MyAffCrypto, then assigned to Encrypt. I encrypted my P_Text and assigned to CiphText.

Secondly, I changed type of CiphText to string because I supposed to save this encryption in text file which is called as CipherTextFile.txt. Then I needed to read this file for decryption, so I used CiphTextFile.read() function and assigned to read_Chipdata. Then, I changed encoding type of this data to utf-8.

The third method is decryption. I needed some key, namely inverse_key, and I used these keys in MyAffCrypto function, then assigned to Decrypt. I have generated inverse of the random number. Before decrypting CiphText, I encoded read_Chipdata again in order to use capital letters and encrypted file without space, then I assigned to CiphText2. Final part of decryption is to decrypt this text and I used Decrypt function for CiphText2. After that I checked this method whether it was right or false.

Finally, I suppose to run brute force tests with different ranking parameters. Therefore , I tested none ranking, chisquare ranking and ranking squared_differences then I printed each result to the file respectively.

User Guide:

1. Write your sentence in plaintext.txt
2. Use Encrypt Function to encrypt your file.
3. Use Decrypt Function to decrypt your encrypted file.
4. If you want to check Decrypt(file) and plaintext are equal, you can use Decrypt() == P_Text
   a. If answer is true, try to use ranking.
   b. Else check steps.
5. Ranking Results
   a. If you want none ranking, just use “;RankedNone”
   b. If you want chisquare ranking, use “RankedChis[:10]”
   c. Else use “RankedSquared[:10]”

Outcomes Report

1. Learned reading from text file, and writing into another text file.
2. Have learned encryption and decryption
3. Have learned to generate random numbers, to take the inverse of these numbers and to use them
4. Learned to check whether the result is true or false.
5. Learned to run a brute force test with different parameters.

Sample Run Results:

Input = Plaintext : Affine Crypto Assignment Two

Encoded Text: AFFINECRYPTOASSIGNMENTTWO

Encrypted File = CiphText: YVVOLGCTUPXAYIIOKLWGLXXQA

Decrypted File = CiphText2: AFFINECRYPTOASSIGNMENTTWO

Ranking = “none” Result:

```
((1, 0), YVVOLGCTUPXAYIIOKLWGLXXQA),
((1, 1), XUUNKFBSTOWZXHHNJKVFKWWPZ),
((1, 2), WTTMJEARSNVYWGGMIJUEJVVOY),
((1, 3), VSSLIDZQRMUXVFFLHITDIUUNX),
```

Ranking = “chisquare” Result:

```
((15, 24), AFFINECRYPTOASSIGNMENTTWO),
((3, 16), UTTIHOEBKRLMUGGIYHCOHLLAM),
((11, 10), GBBYTCEPIRNSGOOYATUCTNNKS),
((21, 20), UFFWHIOVABPEUSSWCHKIHPPGE),
```

Ranking = “squared_differences” Result:

```
((15, 24), AFFINECRYPTOASSIGNMENTTWO)
((17, 21), RAAVETFGDSULRNNVHEXTEUUPL)
((17, 8), ENNIRGSTQFHYEAAIURKGRHHCY),
((21, 2), GRRITUAHMNBQGEEIOTWUTBBSQ)
```