Entropy of Encrypted Data

Jiun-Cheng Jiang
Department of Physics National Tsing Hua University, Hsinchu 30013, Taiwan, Republic of China
(Dated: May 5, 2023)

I. INTRODUCTION

A. Details about important properties in information theory

In information theory, we use Shannon information to weigh the content of a message or data. The Shannon information is defined as

$$I = -\log(p_i)$$

where p_i is the probability of the i-th event, (and in this post, we will use Einstein summation convention.) For example, that the sun rises from the east is always true, so the probability of this message is 1. The Shannon information of this message is 0, which means that this message contains no information. On the other hand, if we know the number of the lottery is "1234567890" where each number has the same probability, the Shannon information of this message is $-\log_{10} 10^{-10} = 10$. In this way, we can use the Shannon information to know how much information a message contains.

However, the rare event is not always the most important, since it comes with a low probability. Or in the case of cryptography, what we want is to make the encrypted message into a ordinary random message. In this case, we can use Shannon entropy to describe the uncertainty, or the randomness, of a message. The Shannon entropy is defined as

$$H = -p_i \log(p_i)$$

where p_i is the probability of the i-th event. And we can see that Shannon entropy is just the expected value of Shannon information.

So far, we introduced two kinds of methods to measure the information of one message. However, in some cases, we want to know the relatedness of two messages. In this case, we can use mutual information to measure the relatedness of two messages.

Before we keep going to know the mutual information, we would like to first introduce Kullback-Leibler divergence as the representation of mutual information. The Kullback-Leibler divergence is defined as

$$D_{KL}(p||q) = p_i \log \frac{p_i}{q_i}$$

where p_i and q_i are the probability of the i-th event in two messages. Note that the Kullback-Leibler divergence is not symmetric, which means that $D_{KL}(p||q) \neq D_{KL}(q||p)$. From the equation, we can see that the more similar the two messages are, the smaller the Kullback-Leibler divergence is. And if the two messages are identical, the Kullback-Leibler divergence is 0.

Then, we can use the Kullback-Leibler divergence to define the mutual information as

$$I(X;Y) = D_{KL}(P(X,Y)||P(X)P(Y))$$

where $P(X,Y)$ is the joint probability of X and Y, and $P(X)P(Y)$ is the product of the marginal probabilities of X and Y. We can see that the mutual information is symmetric, which means that $I(X;Y) = I(Y;X)$.

To make a better picture of the mutual information, we can consider a case that X and Y are independent. In this case, the joint probability of X and Y is the product of the marginal probabilities of X and Y, which means that $I(X;Y) = 0$. On the other hand, if X and Y are identical, the joint probability of X and Y is the same as the marginal probabilities of X and Y, which means that $I(X;Y) = H(X) = H(Y)$. Also, we get the range of mutual information which is $0 \leq I(X;Y) \leq min(H(X), H(Y))$.

Finally, the last piece of pizza, I mean puzzle, is the conditional entropy. Conditional entropy is the entropy of one random variable X when given the value of another random variable. The conditional entropy is defined as

$$H(Y|X) = p(x_i)p(y_j|x_i) \log p(y_j|x_i)$$

where $p(x_i)$ is the probability of the i-th event of X, and $p(y_j|x_i)$ is the probability of the j-th event of Y when given i-th event of X.

Now, we can connect the entropy, the conditional entropy and the mutual information by the following equation

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \quad (1)$$

which is the important relationship that we will use in this project.

B. Encryption

There are three elements in encryption: plaintext, ciphertext, and key. Plaintext is the original message. Ciphertext is the encrypted message. Key is the secret used to encrypt the plaintext. The encryption process is

a function that maps the plaintext and key to the ciphertext. The decryption process is a function that maps the ciphertext and key to the plaintext.

Firstly, we consider the single-bit encryption. To make sure that once we know the ciphertext and the key, we can recover the plaintext, the encryption function must be a bijection. The possible encryption functions are:

$$f_1(0,0) = 0, f_1(0,1) = 1, f_1(1,0) = 1, f_1(1,1) = 0$$

$$f_2(0,0) = 1, f_2(0,1) = 0, f_2(1,0) = 0, f_2(1,1) = 1$$

f_1 is XOR (exclusive-OR) encryption, and f_2 is XNOR (exclusive-NOR) encryption. They both have the characteristic that ciphertext is corresponding to both 0, 1; only once we know the key is, the plane text can be recovered. And in most of cases, we pick XOR encryption as the most basic encryption to bit-wise operation.

Before we take account into multi-bits encryption, there are two important properties of the operation of a secure cipher identified by Shannon: confusion and diffusion. Confusion means that the relationship between the ciphertext and the key must be complex and involved. Diffusion means that the statistical structure of the plaintext must be dissipated into long-range statistics of the ciphertext.

There are two common ways to achieve confusion and diffusion which are Substitution-Permutation Network.

a. SP-Network contains two main features: substitution and permutation. Substitution means that the plaintext is replaced by the ciphertext using a substitution table. Permutation means that the order of the bits in the ciphertext is changed. Finally, at each round, the round key is combined using XOR encryption and sends the results to next round. The SP-Network is illustrated as follows:

b. Feistel Cipher use another way to realize confusion and diffusion. It splits the plaintext into two halves, and each round, the left half is XORed with the round key and then the result is sent to the round function. The output of the round function is XORed with the right half and then the result is sent to the next round. The Feistel Cipher is illustrated as follows:

Based on these concepts, we have many kinds of algorithm to encrypt the data. We have an introduction about 3 of the most common encryption algorithms which is written with the help of GPT: https://github.com/Jim137/Entropy/tree/main/encryption_algorithm#readme. Also the example code can be found in 'src' folder.

All we have discussed above is that the encryption is about bit or binary system. Now, if we have a basis set of the data, e.g.: $\{0,1,2,3,4,5,6,7,8,9\}$ or $\{A,B,C,...,Z\}$, we still can use the index to map the basis set to the binary data, like Base64 encoding. For the N-bases system, we have 11 methods introduced by Shannon in 1949 to encrypt the data, including:

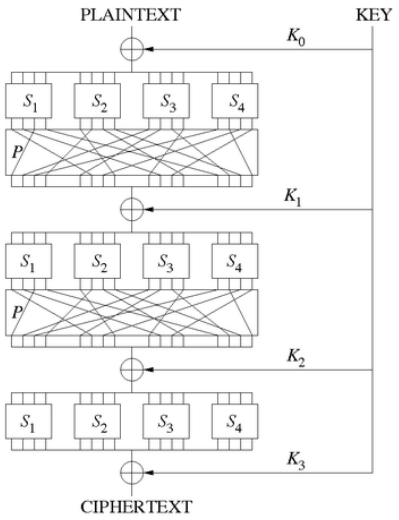

FIG. 1. A sketch of a SP-Network with 3 rounds.[a]

[a] Cited from https://en.wikipedia.org/wiki/Substitution-permutation_network.

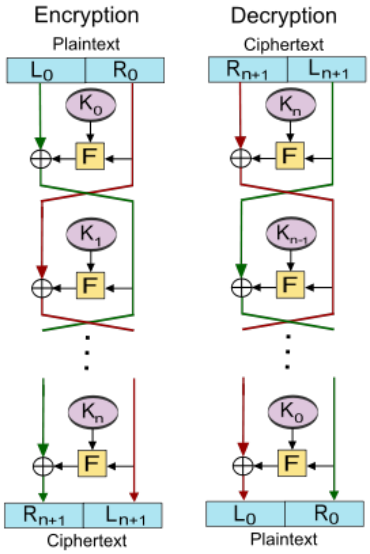

FIG. 2. A sketch of a Feistel Cipher.[a]

[a] Cited from https://en.wikipedia.org/wiki/Feistel_cipher.

- Simple Substitution Cipher
- Transposition
- Vigenère cipher
- N-grams Substitution Cipher
- Single Mixed Alphabet Vigenère Cipher
- Hill Cipher
- Playfair Cipher

- Multiple Mixed Alphabet Substitution
- Autokey Cipher
- Fractional Ciphers
- Codes

To make it clear, we can construct a function mapping the plaintext with the key to ciphertext as shown in the following form:

$$f : \{m_0, m_1, m_2, ..., m_{N-1}\} \times \{m_0, m_1, m_2, ..., m_{N-1}\} \\ \rightarrow \{m_0, m_1, m_2, ..., m_{N-1}\}$$

which f needs to satisfy the following properties: 1. f is a bijection. 2. Ciphertext must correspond to every element in the basis set. That without the key, we cannot recover the plaintext.

Given a 3-elements basis set $\{A, B, C\}$, we have $3!2! = 12$ possible encryption functions, one of them is:

f	A	B	C
A	A	B	C
B	B	C	A
C	C	A	B

And a general form of encryption function can be represented as:

$$c_i = f(p_i, k_i) = a \cdot p_i + b \cdot k_i + c \mod N$$

where a, b, c are integers, N is the number of elements in the basis set, p_i is the plaintext, k_i is the key, and c_i is the ciphertext. And the decryption function is:

$$p_i = f^{-1}(c_i, k_i) = a \cdot c_i + b \cdot k_i + c \mod N$$

where we can see that $f = f^{-1}$, in this case, f is so-called symmetric encryption.

To realized the perfect secrecy in this way, we need to have a one-time pad (OTP), which has been proved mathematically that it can achieve perfect secrecy and quantum resistance. However, it still have restrictions in the practical application, such as pre-shared key, key length, and key distribution *(P.S. Quantum Key Distribution (QKD) may be helpful in OTP.)*

Nowadays, what we usually use in daily communication is the asymmetric encryption (as known as Public-key cryptography), such like RSA, Elliptic Curve Cryptography (ECC) and so on. In this case, we have two keys, one is public key, and the other is private key. The public key is used to encrypt the data, and the private key is used to decrypt the data. The public key is usually published to the public, and the private key is only known by the owner.

The encryption function and decryption function now will no longer be a linear function as symmetric one, but a nonlinear function. The encryption function is usually a trapdoor function, which is easy to compute in one direction, but hard to compute in the opposite direction without the private key. Therefore, we can use asymmetric encryption to exchange the symmetric key, and then use the symmetric encryption to encrypt the data.

However, it has been proved that RSA is not quantum resistant, which means that the quantum computer can break the RSA in polynomial time with Shor's algorithm. And now, we have Post-quantum cryptography (PQC) to resist the attack from both classical and quantum computer.

II. METHODS

We constructed a 4-bits XOR cipher as an example to show how to calculate the entropy of the encrypted data, mutual information and conditional entropy to show the relation of plaintext, key and ciphertext.

In addition, to show the **fractal structure** of XOR encryption, we also constructed 10-bits and 8-bits XOR cipher.

The details of the methods can be found in entropy_of_encrypted.ipynb in my GitHub repository: https://github.com/Jim137/Entropy/blob/main/entropy_of_encrypted.ipynb.

III. RESULTS

*Note: the base of logarithm we used is 2 which represents **bit** in the unit of entropy.*

To start with the 4-bits cipher, the following figure shows the ciphertext as the result of XOR operation between plaintext and key. The color represents the ciphertext (or can directly see from the number on it). We can see that the ciphertext is equally distributed to every element in the basis set. And the result is like Sudoku (數獨) that the number of ciphertext is not showing again on its row and column.

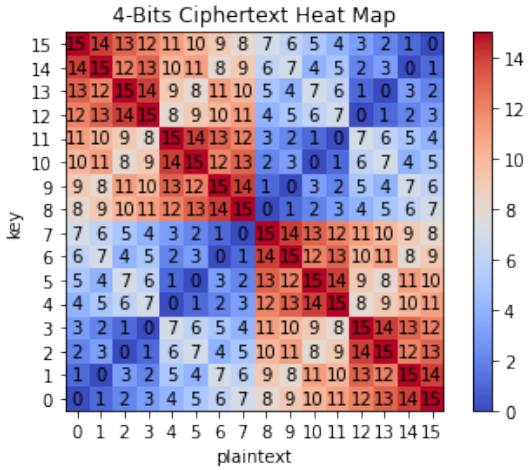

FIG. 3. 4-Bits Ciphertext Heat Map.

And to calculate the properties in information theory later, we firstly made a bar plot of the entropy in 4-bits basis set to have more clear view. In the figure, we can see that when the numbers of 0 and 1 are equal, the entropy reaches the maximum. And the entropy is at the minimum when all the bits are 0 or 1.

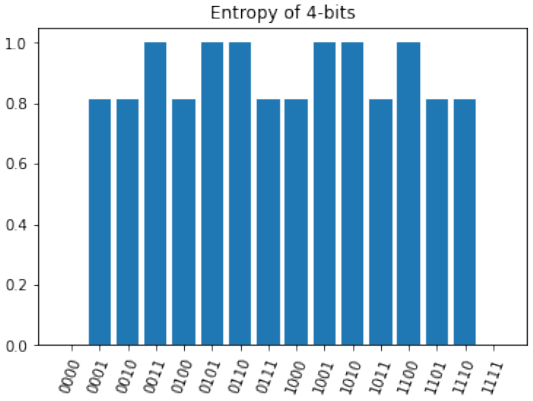

FIG. 4. 4-Bits Entropy Bar Plot.

The entropy of the ciphertext is shown in the following figure. It turns into a four-fold symmetry instead of a fractal structure.

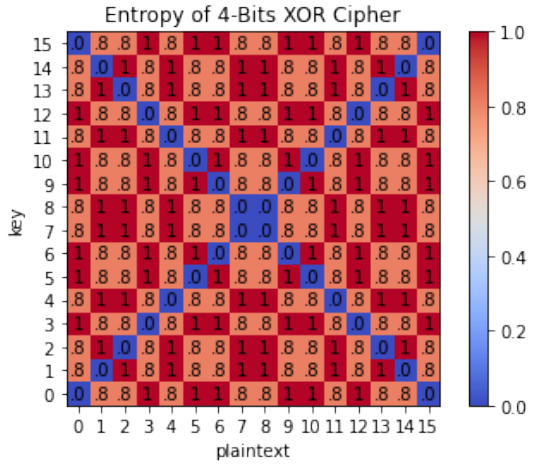

FIG. 5. 4-Bits Ciphertext Entropy Heat Map.

Then the mutual information between C, P and C, K are shown in the followings. We can see that they are a quarter turn to each other. And also, the mutual information between C, P and C, K are small in most of the part, which means that the corelation between the ciphertext and the plaintext or key is weak.

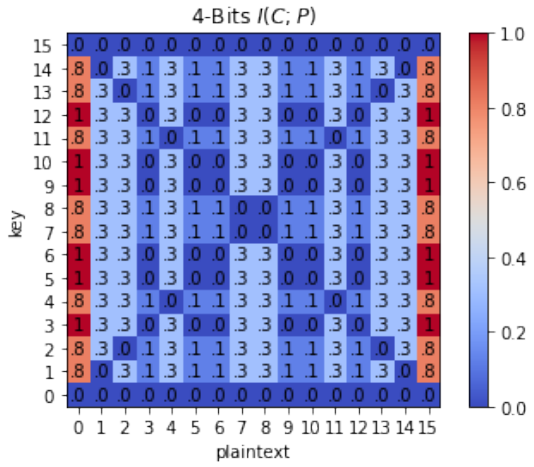

FIG. 6. 4-Bits Mutual Information between C, P and C, K.

With Eqn. 1, we can calculate the conditional entropy in the 4 kinds of relation. Let's see when the plaintext or key is known and how the conditional entropy that ciphertexts have. Based on the mutual information we get, there are cross-like pattern on it and have a quarter turn to each other.

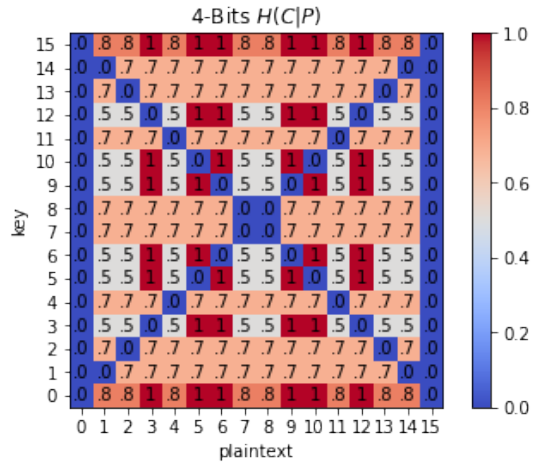

FIG. 7. Conditional Entropy of Ciphertext when Plaintext or Key is Known.

And when the ciphertext is known, the conditional entropy of plaintext and key are shown in the followings. We can see that the conditional entropy of plaintext and key are the same, which means that the plaintext and the key are as well as hard to recover.

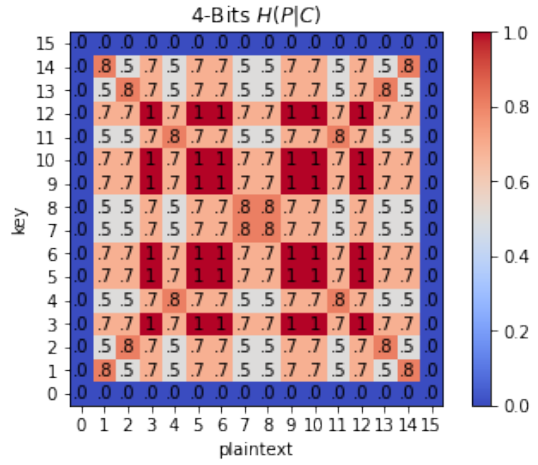

FIG. 8. Conditional Entropy of Plaintext and Key when Ciphertext is Known.

And we can compare the average entropy of 4-bits cipher and 8-bits cipher. We assume that the probability of each states (bits combination) is the same, and we know that the entropy of binary distribution is in the interval of 0 and 1 bit. The result is shown in below table.

	4-Bits	8-Bits
$\langle H_{Basis} \rangle$	0.7806	0.9024
$\langle H_{Cipher} \rangle$	0.7806	0.9024
$\langle H(P\|\|C) \rangle = \langle H(K\|\|C) \rangle$	0.5434	0.7788

TABLE I. Average Entropy of 4-Bits and 8-Bits Cipher.

We found that for equidistributed keys, the average entropy of ciphertexts is same as the average entropy of basis set. However, once we know the ciphertext, the average entropy (conditional entropy) of plaintexts and keys will be lowered down.

That sounds weird, since we encrypt the data, it should be more random and the entropy should be increased. But the result shows that the entropy of plaintext and key are decreased. The reason is that the plaintext should **NOT be equidistributed!** The plaintext we want to send must be meaningful and full of "information".

On the other hand, what we discussed here is single-char encryption (but multi-bits). In real case, we would send a sequence of chars instead of a single char. The average entropy we calculated here only represents the case that if we send a random plaintext with a random key in all the same probability.

Therefore, if we implement the randomly distributed key on the "full-of-info" plaintext, it will make the ciphertext random instead keeping the regularity of plaintext, which results in the increase of the entropy of ciphertext from plaintext.

IV. CONCLUSIONS

Based on the results, we can conclude that the properties of a good encryption are as follows:

a. ***The transformation map from plaintext and key to ciphertext must be equidistributed.*** If the transformation map is not equidistributed, based on char frequency in each languages statistically, the eavesdropper can guess the plaintext from the ciphertext with frequency characteristic.

b. ***The key must be randomly distributed.*** Since we want to make the conditional entropy between ciphertext and plaintext reach the maximum, and the mutual information between ciphertext and plaintext reach the minimum. As a result, the eavesdropper cannot get useful information from ciphertext without key.

c. ***The order of the sequence of chars from plaintext to ciphertext must have a random permutation, and we also have to make the length of ciphertext into a given length instead of original length.*** To satisfy *confusion and diffusion*, it can make the mutual information between ciphertext and plaintext lower, and the conditional entropy between ciphertext and plaintext higher. As a result, it will make the eavesdropper even harder to get useful information in the case that we use the same key and communicate multiple times.

Appendix A: FRACTAL STRUCTURE

In the jupyter notebook of methods, I have presented a special characteristic of XOR which is Bitwise Fractal. However, I cannot make a good interpretation of them,

also it will exceed the topic. So I don't put it in the main text, but I still want to show how cool it is. Therefore, I put it in appendix.

We can clear see that the following figure, 10-bits cipher, presents a fractal pattern.

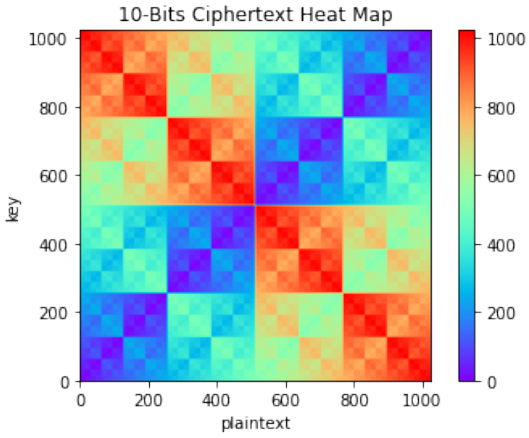

FIG. 9. 10-Bits Ciphertext Heat Map.

The unit structure of the fractal we can see that is XOR, where we prepared a XOR true table below. That the diagonal have higher value than the anti-diagonal. And the above figure did show the XOR behavior in every two-power-series scales' structure.

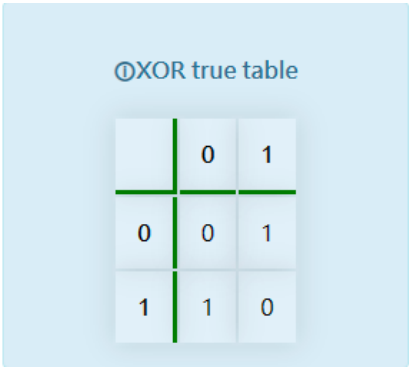

FIG. 10. XOR True Table.

We also can see that they still satisfy the property what we concluded above.

The entropy of cipher turns into four-fold symmetric instead of fractal pattern.

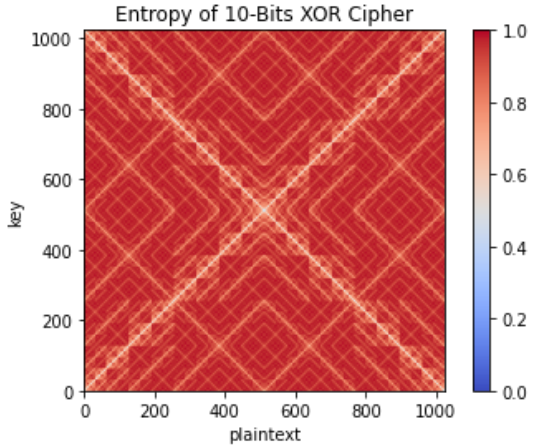

FIG. 11. Entropy of 10-Bits XOR Cipher.

Mutual information plots are a quarter turn to each other.

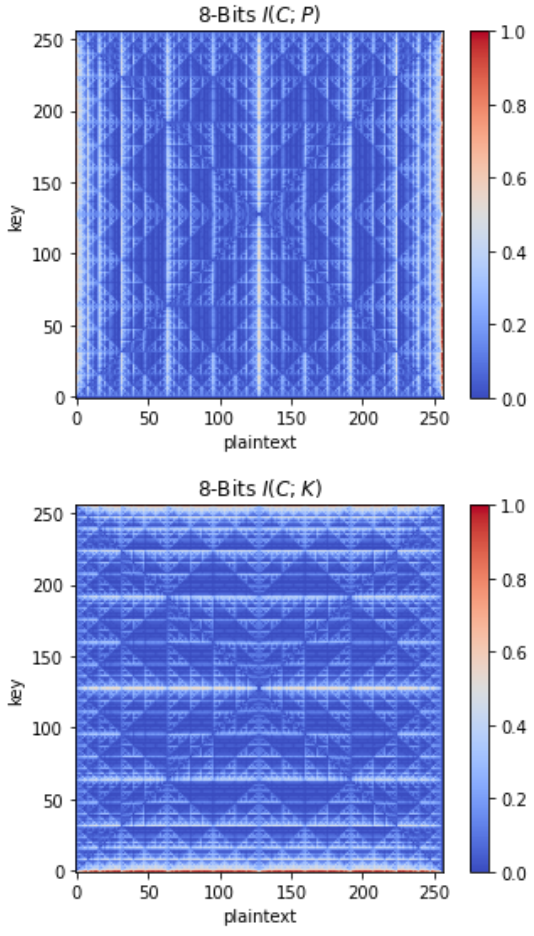

FIG. 12. Mutual Information of 8-Bits Cipher w.r.t. C, P and C, K.

So do conditional entropy plots when plaintext or key is known.

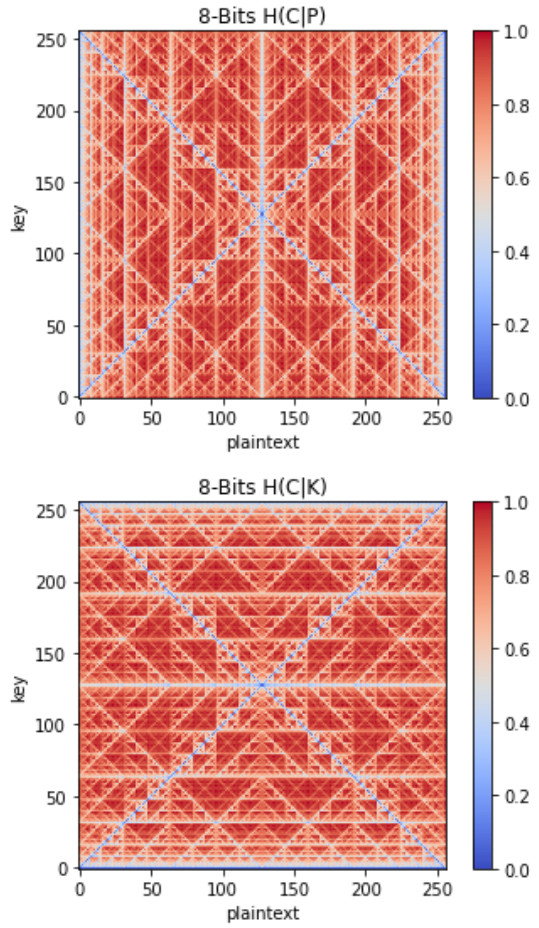

FIG. 13. Conditional Entropy of 8-Bits Ciphertext when Plaintext or Key is Known.

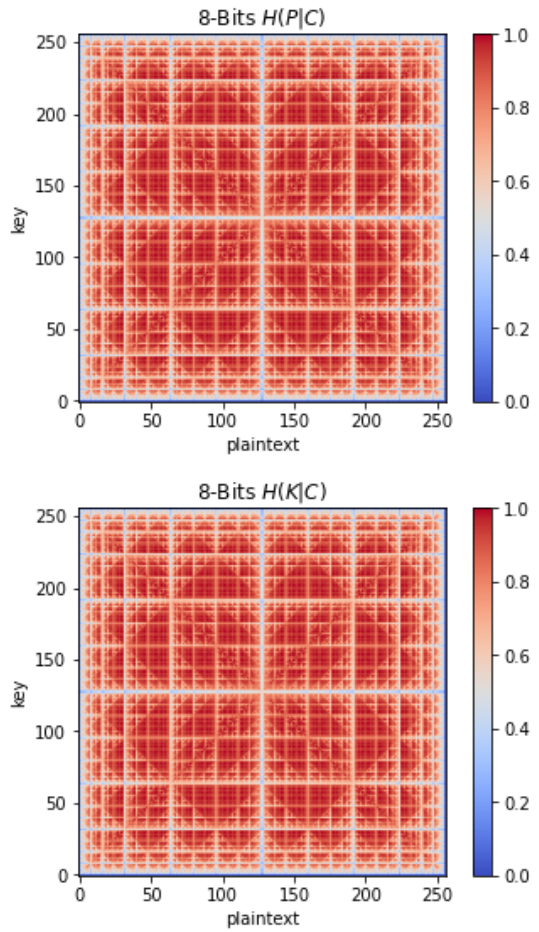

FIG. 14. Mutual Information of 8-Bits Plaint or Key when Ciphertext is Known.

And when ciphertext is known, the conditional entropy plots to plaintext and key are the same.

All the properties above are the same for 4-bits, 8-bits and 10-bits cases. But for higher number of bit, the fractal pattern is more significant. I think this is a cool phenomenon, that I want to introduce.

[1] C. E. Shannon, The Bell System Technical Journal **28**, 656 (1949).

[2] C. E. Shannon, The Bell System Technical Journal **27**, 379 (1948).