**Approaches**

***Vigenère Cipher***

For the Vigenère cipher, cryptanalysis is broken down using the following primary functions:

**vigenere** returns the key and plaintext of a ciphertext encrypted via the Vigenère cipher

**compute\_periods\_with\_ic** returns the period of a polyalphabetic cipher

**Polyalphabetic cipher period**

Since Vigenère cipher uses a form of polyalphabetic substitution, the period needs to be found and analyzed as part of the cryptanalysis. The period is the number of letters encrypted before a polyalphabetic substitution cipher returns to its first cipher alphabet. And to obtain said period, the first step to take is to utilize the Kasiski method. Using the function **kasiski**, its first step is to step into another function, **find\_trigram**, to find the any existing trigrams that are repeated within the ciphertext.

1. def find\_trigram(ctx): ""
2. "Find trigrams within the ciphertext"""
3. tri = {}
4. for i in range(len(ctx) - 3 + 1):
5. curr\_gram = ctx[i: i + 3]
6. tri[curr\_gram] = tri.get(curr\_gram, 0) + 1
7. tri = sorted(tri.items(), key = itemgetter(1), reverse = True)
8. tri = [(a\_gram, occur) for a\_gram, occur in tri if occur > 1]
9. return tri

After finding all of the existing trigrams, the distance between the repeating trigrams is calculated. After calculating the factors, the 5 most frequent are considered as possible periods of the ciphertext, concluding the Kasiski method function.