Vigenere Cipher

Wednesday, September 2, 2020 11:35 AM

Invented in the 1500s (I think)
Surprisingly, a general technique to break it didn't come out until the 1800s.

Plaintext: I T I S S U N N Y O U T S I D E
Keyword: K N I G H T S

8	19	8	18	18	20	13	13	24	14	20	19	18	8	3	4	
10	13	8	6	7	19	18	10	13	8	6	7	19	18	10	13	
18	32	16	24	25	39	31	23	37	22	26	26	37	26	13	17	
	6				13	5		11		0	0	11	0			
S	G	Q	Y	Z	N	F	X	L	W	A	A	L	A	N	R	

c[i] = (p[i] + key[i%keylength])%26 (in numbers)

p[i] = (c[i] - key[i%keylength] + 26)%26 (in numbers for code)

Basically, instead of doing 1 shift cipher, we have different shift ciphers for different letters.

For example,

Letters 0, 7, 14, 21, etc. are all shifted by K
Letters 1, 8, 15, 22, etc, are all shifted by N,
etc.

A good number of possible keys, in theory 26 power of the length of the word, in practice, probably a few million at most, since most are words or close to words.

Different Plaintext letters can map to the same ciphertext letter.
The same plaintext letter can map to two different ciphertext letters.
So this fundamentally messes up frequency analysis.

Frequency information about the plaintext was simply destroyed.

A keyword can be any string you want it to be where each letter can be any of the 26 possible options. (In practice, these weren't random and were usually words or combinations of words.)

Good Exercise: Code up a program that takes the plaintext and the key and gives the encryption using Vigenere (5 to 10 lines long), also do decrypting…

Even though frequency isn't maintained, the frequency of bins is maintained…so if we happened to know that the key word length was 7, then we could put these letters into 7 separate bins:

c[0], c[7], c[14], c[21], etc. (these letters were shifted by the same offset, so these letters should have the same frequency as English)

c[1], c[8], c[15], c[22], etc. (these letters were shifted by the same offset, so these letters should have the same frequency as English)

c[2], c[9], c[16], c[23], and so on…

If we could guess the keyword length, what we ought to see is that these bins have frequency info. The downside is that these bins don't have letter patterns.

How might we obtain the keyword length?

1) Kasiski Test - by chance, every now and then, the same word will appear twice in the plaintext AND will be spaced out by some number of characters, that is the multiple of the keyword length! (Example: "THE" appears four times in the text and the keyword length is 7.

   The T can start in one of 7 places…so the probability that the first T is in a new place is 7/7. Now, consider the second T for the second the, the probability it's in a different place than the first is 6/7. For the third t, the probability it's in a different place than the first 2 is 5/7. For the fourth t, the corresponding probabiliby is 4/7. Probability that all four T's are in different

"slots" so to speak is 7/7 * 6/7 * 5/7 * 4/7 = 0.3499, which means that 65% of the time, at least two of the THEs will line up a multiple of 7 letters apart…And in this case, they will encrypt as the same thing.

It turns out, that if you see a repeated trigram or n-gram (n>= 3), there's a really good chance that actually maps to the same plaintext!!!

So, let's say we see the same ciphertext trigram at indexes 106, 190, 260. Then it means that whatever the keyword length is, it must divide evenly into 190-106=84, and it must ALSO divide evenly into 260-190 = 70. So we should find the gcd…gcd(84, 70) = 14…the keyword length likely divides into 14.

2) Another strategy is to use the index of coincidence test. In English coincidence loosely means when two things turn out to be the same, but by random chance, not any real intention. In mathematics, the definition of index of coincidence is as follows:

Given a multi-set of objects, S, what is the probability, that, if we randomly select two items from S, that they are the same thing?

Example: Bag of Candy - 5 M&Ms, 10 Snickers, 12 Hersheys, and 8 Twix. Kid #1 grabs a candy at random, and so does kid # 2, what is the probability that they both got the same type of candy.

$$\frac{5\times4+10\times9+12\times11+8\times7}{35\times34}$$

242
56
298

$$=\frac{20+90+132+56}{5\times7\times2\times17}$$

$$= \frac{}{5 \times 7 \times 2 \times 17}$$

$$= \frac{298\ 149}{5 \times 7 \times 2 \times 17} = \boxed{\frac{149}{595}}$$

The Index of Coincidence of a random piece of English text is, on average, about .0676.

$$\frac{\sum_{i=0}^{25} f_i(f_i - 1)}{n(n-1)}$$

What might you expect the index of coincidence or a set or random letters to be? 1/26 because the chance you match something random with the second letter is 1 divided by how many letters there are. 1/26 ~ 0.038.

The way we use index of coincidence to find the keyword length is as follows:

Try different values of k (2, 3, 4, …):

For each value of k, split the ciphertext into k bins.
c[0], c[k], c[2*k], c[3*k]
c[1], c[k+1], c[2*k+1], c[3*k+1],…
…
c[k-1], c[2*k-1], c[3*k-1], …
For each of these k bins, take their index of coincidence.
If most bins are close to 6 or 7 percent, then your guess for k is

probably right.

If most bins are around 4 to 5 percent, your guess for k is probably wrong.