Substitution Cipher and its Cryptanalysis

A straight substitution cipher involves substituting each letter for another one. Each letter can be chosen as a substitute exactly once. In essence the ciphertext alphabet is nothing but a permutation of the plaintext alphabet. Thus, there are 26! possible keys for the substitution cipher.

Unlike both Shift and Affine, this keyspace is simply way too big to try all keys.

Thus, we must endeavor to do true cryptanalysis, which is the practice of using information to eliminate some of the possible keys and only search the ones that might actually be the secret key. Thus, our analysis of both the shift and affine cipher, was technically not cryptanalysis, since all we did for both is try all possible keys. The only real cryptanalysis shown so far was using 2 guesses for matching plain and ciphertext characters to yield a system of two equations to decrypt a ciphertext encrypted by the affine cipher.

When players play Wheel of Fortune, or hangman, they tend to guess the same letters first. This is not a coincidence. Rather, we all intuitively know that not all letters appear with the same frequency in regular writing in English. In fact, here is a frequency chart (not sure who generated it, but it's been in multiple textbooks), which shows the percentage of all letters in a set of selected texts that were each of the 26 letters:

Let	A	B	C	D	E	F	G	H	I	J	K	L	M
Frq	.082	.015	.028	.043	.127	.022	.020	.061	.070	.002	.008	.040	.024
Let	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frq	.067	.075	.019	.001	.060	.063	.091	.028	.010	.023	.001	.020	.001

Thus, since we have some baseline information about how often each letter appears in English, **and we know that the Substitution Cipher leaves the set of letter frequencies unchanged**, we can use this information to try more likely possible keys first. For example, if a particular letter, say W, appears in the cipher text 13% of the time, and we know that E appears in English text about 13% of the time, so it's fairly likely that the ciphertext W maps to the plaintext E. And even if it's not E, it's near impossible that the letter maps to 'Z'. In fact, with extremely high probability, it's probably either E, T, A or O.

For generally, here is a chart sorting out the letters by frequency, in groups:

Here are the letters sorted roughly by frequency (in English):

High: E, T, A, O, N, I, R, S, H
Medium: D, L, U, C, M
Low: P, F, Y, W, G, B, V
Rare: J, K, Q, X, Z

Thus, the idea is as follows:

(1) Calculate the frequencies of each letter in the ciphertext.

(2) Look at the top few letters, and either arbitrarily or based on other information about language structure, make some guesses as for substitutions for letters such as ‘E’, ‘T’, and ‘A’.

(3) Once these substitutions are made, look for likely locations for matching letters. For example, if you see “T_E” after substituting T and E, then perhaps that middle letter is ‘H’.

(4) If you continue making substitutions and you end up with some impossible string, then you must have made a mistake. Backtrack and try a different substitution.

(5) Make sure you keep track of which substitutions you’ve tried and which ones you 100% can rule out. It’s easy to go in circles and try the same exact substitution twice without knowing it. Doing so can lead to wasted time. So, any sort of formal accounting system for keep track of what you have tried is extremely important to help you save time.

(6) Beyond frequency information, there are other characteristics of letters in a language that can be exploited. For example, there are common digrams and trigrams in a language.

Here are also a list of common digrams and trigrams (in order by frequency):

Digrams: TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, and OF

Trigrams: THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR and DTH

So, see if once you substitute for a couple letters, if you can make out where these digrams or trigrams might be in the plaintext to fill in extra letters.

(7) Similarly, there are other letters that are unlikely to be adjacent to one another in English. All of these types of properties can be used to narrow down which letters are likely to substitute for which other letters.

Through frequency analysis, nearly all of these possibilities can be quickly discarded. Though it’s tedious, a straight substitution cipher has been reliably broken since about the 10th century by those who knew the methods of frequency analysis and searching for common digrams and trigrams.

In particular, the first recorded instance of the frequency analysis technique dates back to Ismail al Kindi in the 900s. (Code Book)

As mentioned, frequency analysis isn’t perfect. You can’t just do a one to one match for sorted frequencies in English with the sorted frequencies in a ciphertext. First of all, with a small amount of ciphertext, there is lots of “error” in that a small sample of text can easily contain more or less than the long-term average amount of a particular character. (Consider a Wikipedia entry on zebras.) Secondly, on occasion, people write weird things such as an entire novel without the letter E (Code Book).

Thus, the key is flexibility and being willing to make changes to your substitution guesses when other structural clues seem to contradict a particular substitution.

Since it was largely known by many that substitution could be broken, here are some “twists” that codemakers tried to thwart substitution analysis:

1) Created a nomenclature, where some very common words in the plaintext are substituted by symbols. In this system, there are more ciphertext characters than plaintext characters. It strengthens the cipher because cryptanalysts can’t just find multiple repetitions of very common words like “the” and get those letters. The most famous nomenclature is Queen Mary of Scots’. Details about this story are included in The Code Book.

2) The use of null characters that are actually supposed to be ignored by the person decrypting the message. (These can also disrupt frequency analysis.)

3) The use of a delete character, which means to get rid of the previous character. This disrupts frequencies in two ways, but adding a character that doesn’t map to anything in the plaintext (the delete character), and subtracting one from a character that looks to map to a plaintext character but now no longer does.

4) Having the ciphertext contain 100 symbols, 00 through 99, and having 12 map to ‘E’, since 12% of letters are E, etc. In this way, each of the 100 ciphertext symbols appears approximately 1% of the time. When encrypting, the user randomly chooses between all possible ciphertext symbols for each plaintext symbol.

What’s surprising is, as was the case in the Queen Mary of Scot’s cipher, even with these difficulties, codebreakers such as Sir Francis Walsingham were able to break substitution ciphers with all of these bells and whistles. As mentioned in The Code Book, The Queen Mary of Scot’s cipher, in addition to 26 substituted letters, used 36 codewords, 4 null characters and a code character to represent that the following letter was doubled.

Note: The Queen Mary of Scot’s story is pretty cool. You can either read about it in The Code Book or attend lecture =)