Classical and Contemporary Cryptography: Chapter 2 Notes
Monoalphabetic Ciphers

Shift Cipher (not in book, but added for completeness)

The first cryptographic scheme most people either learn or discover on their own is the shift cipher. Rumor has it that Julius Caesar would send messages to his commanders via runners in an encrypted format so that the runners wouldn't get any critical information. Since his runners weren't terribly bright, all he did to encrypt his messages is count three letters forward from the letter he wanted to express. For example, "CAT" would encrypt to "FDW", since F comes three letters after C, D comes three letters after A and W comes three letters after T. The message recipient could easily "subtract 3 letters" to recover the original message, since he "knew the trick", so to speak. Finally, if we get to the end of the alphabet, we simply wrap around. Thus, X encrypts to A, Y encrypts to B and Z encrypts to C.

Mathematically, we need to express each letter as an integer in between 0 and 25 and utilize the mod operator to formally describe the Caesar cipher, which has a shift of three. The encryption and decryption functions are as follows:

$E(x) = (p + 3) \bmod 26$
$D(y) = (c - 3) \bmod 26$

In these formulas, p represents the plaintext (normal) letter and c represents the ciphertext (encrypted) letter.

The mod mathematically deals with the wrap-around at the end of the alphabet. This explains why we use the numbers 0 through 25 instead of 1 through 26. The former choice makes this wrap-around issue much easier to express mathematically.

The Caesar Cipher is a specific instance of the Shift Cipher. In the shift cipher, instead of always shifting over 3 letters, one can shift over any number of letters they choose. This number is known as the key for the shift cipher. Thus, the shift cipher has a total of 26 possible keys, of which one (0) would be completely useless.

In general, the functions for encrypting and decrypting using the shift cipher are as follows:

$E(x) = (p + k) \bmod 26$
$D(y) = (c - k) \bmod 26$

where k, represents the numerical value of the key.

Note: Some books/sources refer to the Caesar cipher as being identical to the shift cipher, so that the term Caesar cipher does not ALWAYS infer a shift key of three.

Cryptanalysis of the Shift Cipher

The easiest way to break the shift cipher is simply to try all 26 shifts until you see a message that makes sense. There are so few keys to try that this can be done by hand relatively quickly. People with good instincts can probably eliminate several choices very quickly. Alternatively, it should be pretty easy to write a computer program that tries all 26 shifts and prints out all 26 possible messages. Here's some C code to accomplish that task that assumes that cipher stores only lowercase letters:

```
void printAllPossibilities(char cipher[]) {

    int i, j;
    for (i=0; i<26; i++) {

        for (j=0; j<strlen(cipher); j++)
            printf("%c", (cipher[j]-'a'+i)%26 + 'a');
        printf("\n");
    }
}
```

In this code, the key difficulty (sorry for the pun!) is that we must convert from characters to integers and back. Given the character cipher[j], we must convert that to its numerical value (in between 0 and 25) by subtracting the ascii value of the character 'a'. Then we need to add to this the shift we are trying, in this case, i. After we use mod to map the value back into the range 0 – 25, we need to then add back the ascii value of 'a' to obtain the new character to print.

Keyword Cipher

A keyword cipher is a specific type of substitution cipher. A standard substitution cipher is where each letter in the plaintext alphabet is replaced with another letter in the plaintext alphabet, just like the cryptograms in the newspaper.

Here is how to implement the keyword cipher:

1) Pick a keyword without repeated letters. (Or if repeats are in the keyword, eliminate all but the first occurrence of each.)

2) Fill out the substitution chart for each letter by placing the keyword first, and then filling in the rest of the letters.

For example, if the keyword is: mushroom, which would be shortened to mushro, then the substitution chart is as follows:

Plain	A	B	C	D	E	F	G	H	I	J	K	L	M
Cipher	**M**	**U**	**S**	**H**	**R**	**O**	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Plain	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher	**I**	**J**	**K**	**L**	**N**	**P**	**Q**	**T**	**V**	**W**	**X**	**Y**	**Z**

Here is an example of implementing this keyword substitution:

```
Plaintext:  WE WENT TO THE SHOW
Ciphertext: WR WRIQ QJ QBR PBJW
```

We can vary the keyword cipher by allowing ourselves to start the keyword at a different starting point than the letter A. Consider picking the letter V:

Plain	A	B	C	D	E	F	G	H	I	J	K	L	M
Cipher	**O**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**I**	**J**	**K**	**L**	**N**
Plain	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher	**P**	**Q**	**T**	**V**	**W**	**X**	**Y**	**Z**	**M**	**U**	**S**	**H**	**R**

Cryptanalysis of Keyword Cipher

Assuming that the keyword was in the dictionary, a brute force search of all the words in the dictionary would conceivably work, if we used a computer. Without a computer this would be quite time-consuming.

In the spirit of cryptanalysis, our goal will be not to use brute force, and to do less work but still recover the key.

The first piece of information to use is frequency of each symbol, since there are one for one substitutions for each symbol, and in the English language (and other languages) each of the letters typically appear at particular frequencies. Thus, 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.

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. 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.

Furthermore, since a keyword cipher is being used, it is likely that several adjacent letters are being substituted by another set of adjacent letters, so once you have a couple letters filled in, it's worth trying the next few adjacent letters, and so on.

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

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

(This list is from Cryptography: Theory and Practice by Stinson page 25)

An example of how to use CAP to utilize the information above will be shown in class.

Affine Cipher

A typical convention will be to set A = 0, B = 1, …, Z = 25, so that we can perform numerical calculations on the plaintext to determine the ciphertext.

The affine cipher works in this manner. It too is a specific type of substitution cipher.

The secret key for the affine cipher are two integers a and b such that gcd(a,26) = 1, (where gcd stands for greatest common divisor), and both a and b are in between 0 and 25, inclusive. Given this key, we encrypt as follows:

$c = ap + b \pmod{26}$

For example, if a = 3, b = 7 were the secret key, then the plaintext L (which is 11) would encrypt to O as follows:

$C = 3(11) + 7 \pmod{26}$
$= 40 \pmod{26}$
$= 14$ (O)

The reason a is restricted is that if gcd(a, 26) > 1, then a valid substitution would not be created. We won't go over the proof of this, but we'll go over a quick example.

If a = 2, and b = 5, then consider encrypting both B(1) and O(14):

$c = 2(1) + 5 \pmod{26} = 7$ (H)
$c = 2(14) + 5 \pmod{26} = 33 \pmod{26} = 7$ (H).

But, it's impossible for both B and O to encrypt to H, because then, if we see an H in the ciphertext, we won't know what the corresponding plaintext was.

There are a total of 12x26 = 312 possible keys for the affine cipher. (There are 12 values in between 0 and 25 that have a gcd of 1 with 26.)

Cryptanalysis of the Affine Cipher

All you need to determine the key for an affine cipher is two matching plaintext-ciphertext characters. Consider the following example:

If through frequency analysis you thought that the plaintext E(4) mapped the ciphertext C(2), and that the plaintext T(19) mapped to the ciphertext F(5), then we could set up the following two equations:

$2 = a(4) + b \pmod{26}$
$5 = a(19) + b \pmod{26}$

Subtracting the top equation from the bottom yields

$3 = 15a \pmod{26}$
$7(3) = 7(15)a \pmod{26}$
$21 = 105a \pmod{26}$
$21 = a \pmod{26}$

The difficult part of the work above was "knowing" to multiply the equation through by 15. 15 is actually 7^{-1} mod 26. Namely, when 7 is multiplied by 15 and considered mod 26, it equals the identity, 1. We won't go through how to determine these inverses in this class, but here's a quick list of them for mod 26:

Value	1	3	5	7	11	17	25
Inverse Mod 26	1	9	21	15	19	23	25

Also, since there are only 312 keys, one can simply search all of these.

2.3 Multiliteral Cipher

To implement the multilateral cipher, pick a code word of length five that has no repeated letters. Consider the code word "CRYPT." Create a 5x5 matrix like so:

	C	**R**	**Y**	**P**	**T**
C	A	B	C	D	E
R	F	G	H	I/J	K
Y	L	M	N	O	P
P	Q	R	S	T	U
T	V	W	X	Y	Z

To encrypt a letter, find it in the matrix and replace it with the row and column letter that corresponds to it. For example, W encrypts to TR, since it's in row T and column R. Both I and J encrypt to RP. When decrypting, context can be used to choose between I and J.

This cipher only has 5! = 120 possibilities once the ciphertext is viewed. All that has to be decided is in which order the five letters are used to form the code word. All different possibilities can be tried. Furthermore, this cipher is also suspect to basic frequency analysis, since each digraph represents a particular letter. This frequency analysis could easily be used to fix the positions of some of the ciphertext letters in the matrix to reduce the number of options that have to be tried.

2.4 Monoalphabetic Ciphers

A standard monoalphabetic cipher is the straight substitution, which 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. 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 10^{th} century by those who knew the methods of frequency analysis and searching for common digrams and trigrams. Here are some of the techniques people tried to make the substitution cipher more difficult to break:

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 other clever characters, that may delete previous characters for example.

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.

Cryptanalysis of the Substitution Cipher

For beginning students, this exercise is perhaps the most tedious and frustrating