MA2609: Ciphers and Number Theory

1. Introduction to ciphers

Suppose that two people wish to exchange a message in secret. (Traditionally in cryptography Alice wished to send a message to Bob.) We would like to have ways of encrypting the message so that no one else can read it. (We may call this unwanted recipient Eve.)

Definition 1.1: A code is a system that converts each word or phrase into a different form, while a cipher depends (in some way) on the individual letters in the message. (Thus a cipher is a special kind of code.) We will only consider ciphers here.

Take the message `meet me at noon`. One way to disguise this would be to replace each letter by the next letter in the alphabet (with `z` followed by `a`). In this way we would get `NFFU NF BU OPPO`. As the word lengths might give some information away, it is common to write this in five letter blocks, so the message would be `NFFUN FBUOP PO`.

Assuming that Alice and Bob have agreed this method in advance, Bob can easily decrypt the message by replacing each letter by the one preceding it in the alphabet.

This is an example of one of the oldest ciphers, the Caesar cipher. In general in the Caesar cipher we agree to shift each letter of the alphabet by a single fixed amount.

Example 1.2: Using the Caesar cipher with shift 3 the word `theory` becomes `WKHRUB`.

To decode we shift by the same amount in the opposite direction.

However, if Eve knows we are using a Caesar cipher then she can easily crack it, as there are only 25 possible (non-trivial) shifts.

Definition 1.3: The original message before encryption is called the plaintext. This is encrypted using a particular method, which may depend on certain choices which we call the key. The final version sent is called the ciphertext.

In our example the system was the Caesar cipher, and the key was 3.

We would like to have ciphers which are hard to crack. In general there are several levels of security we might aim for:

Level 1: The cipher cannot be cracked if Eve does not know the system used.

Level 2: The cipher cannot be cracked if Eve knows the system but not the key.

Level 3: The cipher cannot be cracked even if Eve knows the system and the key.

Clearly we also require that Bob can decipher the given ciphertext!

Example 1.4: The Caesar cipher failed to be level 2, as Eve only has to try 25 possible keys. In fact (for suitably long messages) it fails even to be level 1, as we shall see.

Level 3 may look impossible. Surely if Eve knows all the information that Alice used to encrypt the message, then she must also be able to decrypt it? (Otherwise how doe Bob decrypt it?) However we will see that number theory can provide ciphers which satisfy all these levels of security.

Before introducing the necessary mathematics, we will briefly consider a few classical ciphers.

Definition 1.5: The Caesar cipher is an example of a monoalphabetic cipher. In general this is a cipher where each letter is replaced by a single fixed symbol (usually another letter).

Example 1.6: We often write a monoalphabetic cipher in two lines, with the plaintext version on top. For example

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*
1	*C*	7	*L*	!	*G*	*H*	*I*	*X*	*M*	8	*N*	*A*

n	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
B	?	*F*	4	*Q*	*S*	*D*	*R*	*J*	*O*	*P*	*E*	*K*

would encode `apple` as `1FFN!`.

To decode such a message, Bob just has to know the 26 symbols and the letters to which they have been assigned.

There are many possible monoalphabetic ciphers. For practical purposes, when encoding we often want the method to be as simple as possible. For example, if a spy is sending a message from another country then there is a chance that she will be caught. If she needs a written record of the key (for example a list of all 26 symbols) then the enemy would be able to decode all her earlier messages.

With the Caesar cipher Alice only needed to remember one thing, the shift. Other monoalphabetic ciphers can be constructed with simple keys.

The keyword cipher is formed by choosing a secret keyword or keyphrase with no repeated letters. The substitution is then carried out by using the keyword followed by the remaining letters in alphabetic order, as in the following example.

Example 1.7: Suppose the keyword is DUMBWAITER. Then we encrypt using

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*
D	*U*	*M*	*B*	*W*	*A*	*I*	*T*	*E*	*R*	*C*	*F*	*G*

n	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
H	*J*	*K*	*L*	*N*	*O*	*P*	*Q*	*S*	*V*	*X*	*Y*	*Z*

So `meet me at noon` becomes `GWWPG WDPHJ JH`.

To decrypt we use the same table in reverse.

This cipher is a little better than the Caesar cipher as there are now many more possible keys. However it is still easy to crack. Indeed, as with all monoalphabetic ciphers, it does not even satisfy our first level of security.

Before explaining why this is so we will consider another easy class of ciphers, called permutation ciphers.

A permutation cipher takes the letters of the plaintext and rearranges them according to some predetermined pattern. One example is the railfence cipher. Pick some number n of rows, and write the plaintext in zigzag form, then read the ciphertext along the rows.

Example 1.8: To encrypt `this is a railfence encryption` using the railfence cipher with $n = 3$ we write

```
t   i   a   e   e   y   o
h s s r i f n e n r p i n
 i   a   l   c   c   t
```

to get `TIAEE YOHSS RIFNE NRPIN IALCC T`.

To decrypt we have to take the message and reconstruct the zigzag form. If there are n rows then approximately $\frac{1}{2n-2}$ letters occur in the top row, $\frac{1}{2n-2}$ letters occur in the bottom row, and $\frac{2}{2n-2}$ letters occur in the middle rows. This is only approximate, as the total number of letters might not be divisible by $2n-2$. If this is the case, then the extra letters are allocated to the rows in the order:

$$\text{row 1, row 2, } \ldots \text{, row } n \text{, row } n-1, \ldots \text{ row 4, row 3.}$$

This is most easily understood in an example.

Example 1.9: In our message above we took $n = 3$. So approximately $\frac{1}{4}$ of the letters are in row 1, $\frac{1}{4}$ are in row 3, and $\frac{2}{4}$ are in row 2.

There are 26 letters in all, which leaves remainder 2 when divided by 4. Thus:

$$\begin{gathered}\text{line 1 has } \tfrac{24}{4} + 1 \text{ letters} \\ \text{line 2 has } \tfrac{2\times 24}{4} + 1 \text{ letters} \\ \text{line 3 has } \tfrac{24}{4} + 0 \text{ letters}\end{gathered}$$

So we take the first 7 letters (`TIAEEYO`) for line 1, the next 13 letters (`HSSRIFNENRPIN`) for line 2, and the last 6 letters (`IALCCT`) for line 3. Now write the three lines so that the text forms a zigzag as before, and the message can be read off along the diagonals.

There are many possible permutation ciphers, but in general they are all relatively easy to crack if the method of encryption is known.

It is quite easy to detect that a monoalphabetic or substitution cipher has been used (providing the message is not too short). Any cipher is hard to crack given a single short message, but we want ciphers that remain secure even if the messages are long.

We will see in the next lecture how basic statistical methods can be used to detect these two types of ciphers, and to crack the monoalphabetic case.

Given a message encrypted with an unknown cipher, how can it be decrypted? There are various methods that can be used, depending on which kind of cipher is suspected. We will concentrate on a simple statistical approach which is very effective against monoalphabetic ciphers.

This method is know as frequency analysis. It is not guaranteed to work, but is almost certain to be successful (against monoalphabetic ciphers) if the message is long enough.

The basic idea is simple. In English (as in any language that uses letters) some letters are more common than others. By analysing large quantities of text it is easy to generate statistics for the relative frequency of each letter.

In English, the most common letter is E. More precisely, we can expect to see the following frequencies:

E	12.7%	O	7.5%	S	6.3%
T	9.0%	I	7.0%	H	6.1%
A	8.2%	N	6.7%	R	6.0%

with the remaining letters decreasing from 4.3% to less than 0.1% for Q and Z.

Thus if our ciphertext contains approximately 12% of Es, 9% of Ts etc., then it may well be a permutation cipher. If instead it contains some other letter with frequency 12%, followed by letters with frequencies 9%, 8%, 7%, 7% etc., it may well be a monoalphabetic cipher.

These are only *averages*, and in our original plaintext the frequencies may be slightly different, and the order of popularity may vary slightly from that above.

There are other statistics we may use. For example, we may consider which *pairs* of letters are most common. These are called bigrams, or in some books digraphs.

The most common bigram in English is TH. Other common bigrams include ER, HE, IN, AN, ON, RE, ED.

The most common repeated letters are SS, EE, TT, FF, LL, MM, OO.

Using these statistics (and other similar ones) we can usually crack any monoalphabetic cipher. However, some *guesswork* and *trial and error* will usually be involved. Once a few letters have been discovered, we usually need to examine the text itself for clues to help us decipher the remainder.

As an example consider the text in the handout.

We have seen that monoalphabetic ciphers are susceptible to frequency analysis. In this lecture we will briefly consider some alternatives which avoid this problem.

In a polyalphabetic substitution, as the name suggests, we use several different substitutions instead of just one. A simple example is the Vigneère cipher. Pick a keyword or phrase, and write this repeatedly above the plaintext. Now use a Caesar cipher for each letter where the shift corresponds to the keyletter above.

Example 1.10: Suppose the keyword is CABBAGE. To encode `this is a vigenere encryption` we write

c	a	b	b	a	g	e	c	a	b	b	a	g	e	c
3	1	2	2	1	7	5	3	1	2	2	1	7	5	3
t	h	i	s	i	s	a	v	i	g	e	n	e	r	e
W	I	K	U	J	Z	F	Y	J	I	G	O	L	W	H

a	b	b	a	g	e	c	a	b	b
1	2	2	1	7	5	3	1	2	2
e	n	c	r	y	p	t	i	o	n
F	P	E	S	F	U	W	J	Q	P

To give `WIKUJ ZFYJI GOLWH FPESF UWJQP`.

To decode just write the keyword repeatedly above the ciphertext and use the opposite shift. (This is a slight modification of the original Vigneère cipher.)

This method is not susceptible to basic frequency analysis, as the same letter can have different encryptions. However, if it is suspected that the Vigneère cipher has been used then Eve can try frequency analysis using every nth letter (for various possible n) until the frequencies match those of the language of the plain text.

In this example, if Eve had tried $n = 7$ (and had a long enough ciphertext) she would have been able to determine the keyword and plaintext quite easily.

The Playfair cipher is rather different from the others we have seen so far. Choose a keyword or phrase, and write this in rows as the start of a 5×5 grid, omitting repeats of letters, and regarding I and J as the same letter. Then fill the rest of the rows with the other letters in alphabetical order.

Example 1.11: If the keyword is DUMBWAITERS then we get

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

Now take the plaintext and split it into pairs of letters. If a pair is a repeat (eg `aa`) then split it into two pairs by inserting x (eg as `ax ax`). If a single letter is left over, add an x to make the final pair.

Next take each pair and encode as follows:
(i) If the two letters appear in the same row, replace each letter by the letter on its right (cycling round to the beginning of the row if necessary).
(ii) If the two letters appear in the same column, replace each letter by the letter below (cycling round to the top if necessary)
(iii) All other pairs form two opposite corners of some rectangle. Replace each letter by the letter in the same row in the other corner.

Example 1.12: Using the square in Example 1.11 we encode `this is a playfair encryption`.

t	h	i	s	i	s	a	p	l	a	y	f	a	i
R	F	A	C	A	C	R	K	K	I	X	G	I	T

r	e	n	c	r	y	p	t	i	o	n	x
A	R	L	F	E	Z	N	R	E	L	X	M

So the ciphertext is `RFACA CRKKI XGITA RLFEZ NRELX M`.

To decode we reverse the procedure.

This cipher encodes pairs of letters so is not susceptible to basic frequency analysis. However, Eve could use the distribution of bigrams (and other methods) to crack the cipher.

A special case of the Vigneère cipher is the one time pad. Here the keyword is a random sequence of letters as long as the message. This is impossible to break, but needs an enormous keyword (which must be kept somewhere, so could be copied).

We could consider other ciphers of increasing complexity. However we will now turn our attention to finding a *mathematical* way to construct ciphers.