Breaking the Code

"To keep your secret is wisdom; but to expect others to keep it is folly."

Samuel Johnson

"Secrets are made to be found out with time"

Charles Sanford

Codes have been used by the military to keep secrets from the enemy for thousands of years. In the information age, when many people use the internet for banking and shopping, they do not want the information they enter to be available to unauthorised people so such information is encrypted on secure websites.

This report looks at some of the codes which have been used over time and how mathematics is used in making and breaking them.

The Caesar Shift Cipher

The first documented use of codes for military purposes was by Julius Caesar (100-44BC). The type of code he used is commonly known as a Caesar shift. In the following table the middle row gives plain text and the bottom row gives the corresponding cipher text for a Caesar shift of 2 places. It is conventional to write uncoded plain text using lower case letters and coded text using upper case.

Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Code	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

This code can be represented using the mapping $x \to x + 2$ where x stands for the position of the letter in the alphabet. So k, with postion number 10, becomes M with position number 12. Towards the end of the alphabet, y would go to the letter with position number 24 + 2 = 26 but 25 is the maximum position number as there are only 26 letters in the alphabet and we used 0 for a. 26 is subtracted to give 0. This is called addition modulo 26.

Positions 0 to 25 have been used (rather than 1 to 26) because another way to think of modular arithmetic is as a remainder from division. 26 is equivalent to 0 modulo 26 because $26 \div 26 = 1 \text{ rem } 0$.

Using this code, the second quote at the start of this report would be encoded as follows:

secretsaremadetobefoundoutwithtime
UGETGVUCTGOCFGVQDGHQWPFQWVYKVJVKOG

Spaces between words have been removed to make it more difficult to break the code but if someone found this message and knew that it was a Caesar shift code it would not take long to break. There are only 25 possible Caesar shifts; 26 if we include the one that does not move the letters at all but that one is not very secret!

It is possible to write out all 25 possibilities using nothing more complicated than pen and paper and once you have a message that makes sense you know you have decoded it. Using a computer to do all 25 possible decodings would make the process of decoding very quick.

Substitution ciphers

In a substitution cipher each letter is replaced by another letter. A Caesar shift is one type of substitution cipher but the pattern of substitutions in a Caesar shift makes it easy to break. A general subsitution cipher can be made by rearranging the 26 letters of the alphabet and writing this random permutation below the ordinary alphabet. There would be $26! \approx 4 \times 10^{26}$ possible codes. It would take a long time to go through all the possible decodings to find the right one! However, some of the possible substitution codes are too easy to break; the one where each letter is encoded by the same letter is no use but nor are those where a large number of letters do not change.

Permutations where nothing stays in its original place are called derangements. The number of derangements of 26 letters is the subfactorial of 26, denoted !26. To see how to work this out, consider a much shorter alphabet of 4 letters. The total number of permutations is 4! = 24. These are shown below.

abcd	abdc	acbd	acdb	adbc	adcb
bacd	**badc**	bcad	**bcda**	**bdac**	bdca
cabd	**cadb**	cbad	cbda	**cdab**	**cdba**
dabc	dacb	dbac	dbca	**dcab**	**dcba**

The 9 derangements are shown in bold, with a box round them. To calculate the number of derangements, start with the total number of permutations and take away the ones which have at least one letter in its original position.

There are 6 arrangements with a in its original position, these are on a grey background. This is because, with a in its original position, there are 3 other letters which can be arranged in 3! = 6 ways. Likewise, there are 3! permutations with b in its original place and the same for c and d. However, subtracting $4 \times 3!$ will be too much as arrangements with both a and b in their original places have been counted twice.

So $4! - 4 \times 3!$ (1.1) is too small.

If a and b are both in their places, there are 2 letters left which can be arranged in 2! ways. The same is true for other pairs of letters. There are ${}^4C_2$ pairs of letters and so we need to add on ${}^4C_2 \times 2!$ to (1.1) to give

$4! - 4 \times 3! + {}^4C_2 \times 2!$ (1.2)

However, arrangements which have 3 letters in the same places have been counted more than once. In fact, it is not possible for 3 letters to be in the same place and the 4th letter to be in a different place so it is only the one arrangement with all four letters in the same place that has been counted too many times. abcd has been counted 4 times but it should only be counted once.

The number of derangements is

$4!-4\times 3!+{}^4C_2\times 2!-4+1$ (1.3)

This can be written as

$$!4=4!-\frac{4!}{1!}+\frac{4!}{2!}-\frac{4!}{3!}+\frac{4!}{4!}=4!\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\frac{1}{4!}\right) \quad (1.4)$$

In general

$$!n=n!\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+.....+(-1)^n\frac{1}{n!}\right) \quad (1.5)$$

The Maclaurin series for e^x is

$$e^x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+.....+\frac{x^r}{r!}+.....$$

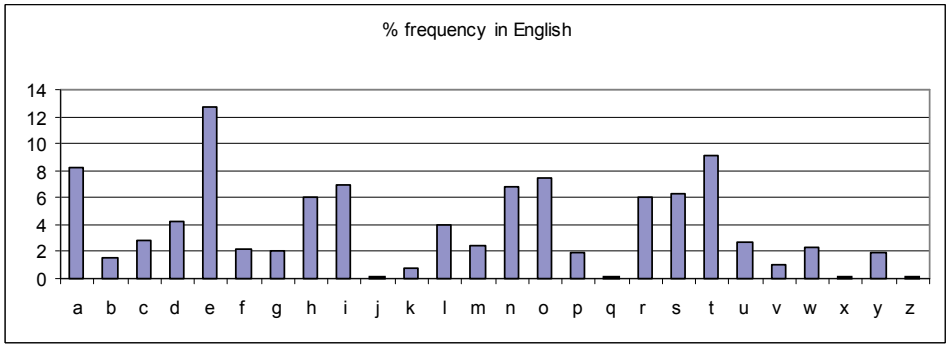
so $1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+.....+(-1)^n\frac{1}{n!}\approx e^{-1}$

$$!n\approx\frac{n!}{e}$$

In fact, rounding $\frac{n!}{e}$ gives $!n$ so the number of derangements of the alphabet is $!26\approx\frac{26!}{e}\approx 1.5\times 10^{26}$. This is still far more possibilities than could be run through in a reasonable time but substitution codes can be broken quickly and easily.

Breaking a substitution cipher

In English, some letters are used more often than others. The following graph shows how often the different letters are used.



If a substitution code is used then the letter that has been used to encode e will be the most frequent letter. Using computers, counting frequencies of letters is quick and accurate so substitution codes are not secure. The earliest known description of frequency analysis for breaking codes dates from the 9th century and is by the Arabic

scholar Al-Kindi. In addition to looking at frequencies of single letters, the frequencies of pairs of letters can be compared with commonly occuring pairs such as th, er, on and so on.

The Vigenère Cipher

In the 16th century Blaise de Vigenère built on the work of others to develop a cipher which was not vulnerable to frequency analysis but which was simple to use. It makes use of Caesar shifts but does not use the same shift all the time. To use the cipher, a keyword is needed. The person doing the coding and the person receiving the message need to know the keyword but that is all they need to remember. By contrast, for a substitiution cipher which is not a Caesar shift, the whole table for coding needs to be either remembered or written down.

The method is illustrated using the keyword STELLA. Caesar shift alphabets which start with S, T, E, L and A will be used.

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Code	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Code	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
Code	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
Code	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
Code	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

To code the initial quote from this report, the keyword is written repeatedly above the text to be coded:

```
stellastellastellastellastellastellastellastellastellast
tokeepyoursecretiswisdombuttoexpectotherstokeepitisfolly
```

The first letter of the message is encoded using the Caesar alphabet which starts with S so it becomes L, the second letter of the message is encoded using the alphabet starting with T so it becomes H. The encoded message is as follows:

LHOPPPQHYCDEUKIETSOBWOZMTNXEZEPIINEOLAICDTGDIPAILBWQZLDR

The Vigenère cipher is not susceptible to breaking by frequency analysis as each letter is coded in different ways. However, it was not widely used because to code each message manually is time consuming as you have to choose the alphabet to use and then find the appropriate encoded letter. Mechanising the process would make it much easier but the use of technology was also what led to the code being broken around 400 years after it was first developed.

Breaking the Vigenère cipher

Charles Babbage, who developed an early computer, realised that to break the Vigenère code the first thing to do was to decide the length of the keyword.

Consider the following encoded text:

JERFIWFLXFIWFLXVHXEAMCNWVHXHIWFLX

Repeated sets of letters in the encoded message are shown in red. They are likely to come from the same letters in the original message being encoded in the same way

and, in this case, it is likely that the keyword is 3 letters long because the distance between repeats is a multiple of 3.

```
1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
J E R F I W F L X F I W F L X V H X E A M C N W V H X H I W F L X
```

If the code word has 3 letters then all the letters numbered 1 above have been coded with the same Caesar shift. The same applies for letters numbered 2 and letters numbered 3. Frequency analysis can now be used separately for each set of letters with the same number to break the code. There is a useful computerised tool for breaking Vigenère codes at http://www.simonsingh.net/The_Black_Chamber/cracking_tool.html

The Enigma Machine

To make adaptations of the Vigenère cipher more secure, a longer keyword needs to be used and it should not be a real word so that guesswork cannot be used in working it out. This makes it much harder to use the Vigenère cipher unless the process can be automated. In 1918, a German inventor, Arthur Scherbius, invented the Enigma machine. It was further developed and used by the German army during the second world war.

The Enigma machine had a keyboard. Operators typed in a message and the machine encoded it. The basic idea is that a rotor is used to move on and encode the next letter using a different substitution alphabet. With one rotor and 26 positions, this would give 26 possible initial starting postions and so 26 possible codes. However 3 rotors were used and each was set to a different starting position every day giving $26 \times 26 \times 26 = 17\,576$ possible codes.

Operators also chose 3 rotors from 5 possible rotors, ie ${}^5P_3 = 60$ ways of positioning the rotors so this now gives $60 \times 17\,576 = 1\,054\,560$ possible codes.

In addition to the 3 rotors, pairs of letters could be connected on the keyboard. If a was connected to b, this would swap the encryption of a with the encryption of b. 10 connections were made, choosing 10 pairs of letters from 26.

The number of ways of choosing the first pair then the second pair and so on is

$$
\begin{aligned}
{}^{26}C_2 \times {}^{24}C_2 \times {}^{22}C_2 \times .... \times {}^{8}C_2 &= \frac{26!}{24! \times 2!} \times \frac{24!}{22! \times 2!} \times \frac{22!}{20! \times 2!} \times ..... \times \frac{8!}{6! \times 2!} \\
&= \frac{26!}{6! \times 2^{10}}
\end{aligned}
\qquad (1.6)
$$

However, the order of selecting the pairs does not matter so each set of 10 pairs has been counted too many times; the answer in (1.6) needs to be divided by 10!

$$\frac{26!}{6! \times 2^{10} \times 10!} \approx 1.5 \times 10^{14}$$

Taken together with the options for the rotors, this gives a total of $1.5 \times 10^{14} \times 1\,054\,560 \approx 1.59 \times 10^{20}$ possible codes. With no possibility of using frequency analysis, the Enigma code was considered to be unbreakable.

Breaking the Enigma Code

Operators of Enigma machines were issued with books which gave them the settings for the day. They were told which letters to connect on the keyboard, which rotors to use and the initial setting for the rotors. If the code breakers had an Enigma machine and could find out the settings for the day, they would be able to decipher all the messages for that day.

Messages would start with the Enigma machine in this position and then the setting of the rotors for the message would be transmitted; the rotors were then adjusted before transmitting the rest of the message. This meant that even if the code breakers had all the messages for the day, each one would use a different code (apart from the first 3 letters of each message). This did not give the code breakers very much to go on.

To break the Enigma code, the code breakers used a copy of an Enigma machine and constructed machines, called bombes, which could work through a large number of possible codes. The machines could not work through anything like all possible codes in one day so further information and deduction was necessary.

The Enigma machine never encoded a letter as itself. This helped with the decoding of some messages. One day a message came through which had no letter L in it. The operator had been sending a test message and had just pressed the letter L every time so it was the only letter that did not feature in the encoded message. Knowing that messages which were sent at particular times of day were about the weather enabled the code breakers to guess part of the message. Knowing that a letter could not be encoded as itself, helped them narrow down the possibilities for which of the possible codes was being used that day and the bombes could then search for that day's settings.

Suppose the word "weather" is known to be part of the plain text for the following coded message. The word can be tried in different positions, as follows

Coded message: QZWPM ZHVVG YGQOZ UQZX

Positioning plain text:

```
weath er
 weat her
  wea ther
```

The third of these possible positions cannot be the place where the word "weather" was in the text because w cannot code to W and h cannot code to H.

Public key cryptography

All the above systems of encoding rely on the key being kept secret from anyone unauthorised who might try to decode the message. The key might be the shift used for a Caesar shift, the keyword for a Vigenère cipher or the initial settings for Enigma machine but once someone knows it they can decode a message.

In the 1970s, Rivest, Shamir and Adleman came up with a way to encode messages which did not rely on the key being kept secret. The system they invented is sometimes known as RSA. The system encodes numbers rather than letters but it is possible to have a system for making letters or words into numbers. The table below shows the steps in the process by using a simple example.

Step	Simple example
Choose 2 prime numbers, p and q. For security, these need to be large.	$p = 2,\ q = 11$
$m = pq$	$m = 22$
Work out $A = (p-1)(q-1)$ and choose a number E which is smaller than this and has no factors in common with it.	$A = (p-1)(q-1) = 1\times 10 = 10$ $E = 3$
Find a number D such that $DE - 1$ is a multiple of A	$3D - 1$ must be a muliple of 10 $D = 7$
To encipher a number M, work out $C = M^E \pmod{m}$	Examples are given below
To decipher a number C, work out $M = C^D \pmod{m}$	
E and m need not be kept secret as long as D is kept secret.	

Enciphering

M	C
1	$1^3 \pmod{22} = 1$
2	$2^3 \pmod{22} = 8 \pmod{22} = 8$
3	$3^3 \pmod{22} = 27 \pmod{22} = 5$
4	$4^3 \pmod{22} = 64 \pmod{22} = 20$

Remember from the section on Caesar ciphers that to find 64 (mod 22) you can find the remainder when 64 is divided by 22. However, working with powers of numbers can lead to calculations with very large numbers. Fortunately mathematics can help.

Suppose you want to work out ab (modulo 15). Suppose $a = r \pmod{15}$ and $b = R \pmod{15}$.

So $a = 15x + r$ and $b = 15y + R$ where x and y are integers.

$ab = (15x + r)(15y + R) = 15^2 xy + 15xR + 15r + rR$

The remainder when dividing ab by 15 is the same as the remainder when dividing rR by 15. So to find what a large number is modulo 15, it is easier to write the large number as a product of numbers and find each of those modulo 15 and then multiply the answers. This can be summed up as follows:

$ab \pmod{m} = ((a \bmod m)(b \bmod m)) \bmod m$.

So $15^3 \pmod{22} = (15^2 \times 15) \pmod{22} = (225 \bmod 22) \times 15 \pmod{22}$
$= (5 \times 15) \bmod 22 = 75 \bmod 22 = 9$

Deciphering

M	C
1	$1^7 \pmod{22} = 1$
8	$8^7 \pmod{22} = (8^2 \bmod 22) \times (8^2 \bmod 22) \times (8^2 \bmod 22) \times (8 \bmod 22) \pmod{22}$ $= ((64 \bmod 22)^3 \times 8) \bmod 22$ $= (20^3 \times 8) \bmod 22$ $= 64\,000 \bmod 22$ $= 2$
5	$5^7 \pmod{22} = (5^2 \bmod 22) \times (5^2 \bmod 22) \times (5^2 \bmod 22) \times (5 \bmod 22) \pmod{22}$ $= ((3^3 \bmod 22) \times 5) \bmod 22$ $= ((27 \bmod 22) \times 5) \bmod 22$ $= (5 \times 5) \bmod 22$ $= 3$
20	$20^7 \equiv 20^2 \times 20^2 \times 20^2 \times 20$ $\equiv 4 \times 4 \times 4 \times 20$ $\equiv 64 \times 20$ $\equiv 20 \times 20$ $\equiv 400 \equiv 4$

To keep writing "mod 22" is making this look more complicated than necessary so, for deciphering 20, a different notation has been used. ≡ means "has the same remainder when divided by 22".

The tables above demonstrate that deciphering takes you back to the original number for the particular examples tested. This will always work. The proof depends on Euler's theorem, which states that

$$M^{\varphi(m)} \equiv 1 \bmod m$$

$\varphi(m)$ is the number of integers smaller than m and having no common factor with with m. If $m = pq$ where p and q are prime, $\phi(m) = (p-1)(q-1)$ and this is what we called A in the description of the RSA algorithm.

So Euler's theorem implies that $M^A \equiv 1 \bmod m$

Using ≡ to mean "has the same remainder when divided by m" and remembering the formula for deciphering is $M = C^D \pmod{m}$

$C \equiv M^E$

$C^D \equiv (M^E)^D$

$\equiv M^{ED} = M^{ED-1} \times M$

$ED - 1$ is a multiple of A so $ED - 1 = kA$ where k is an integer.

So

$C^D \equiv M^{kA} \times M = (M^A)^k \times M$

$\equiv 1^k \times M = M$

Breaking the RSA code

$m = pq$ is known; to break the code needs the individual primes, p and q to be known. $m = 22$ does not give a secure code as it is obvious that the primes were 2 and 11. The primes need to be much larger for the code to be secure.

655 427 is the product of two primes. To find the primes a spreadsheet could be used. This is the start of a spreadsheet showing one way to find the primes:

n	655427/n	Integer part of 655427/n	Factor?
1	655427	655427	yes
2	327713.5	327713	no
3	218475.6667	218475	no
4	163856.75	163856	no
5	131085.4	131085	no
6	109237.8333	109237	no
7	93632.42857	93632	no

Rather than just trying to divide by primes, all integers are tested so 1 does go into 655 427 but this does not help. The prime factors are found further down:

438	1496.408676	1496	no
439	1493	1493	yes
440	1489.606818	1489	no

Constructing the spreadsheet and finding the factors took about 1 minute so 655 427 would not give a secure code.

The factorising can be speeded up by using Fermat's factorising method. This depends on the following result.

$$m = pq = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2$$

Multiplying out the brackets on the right hand side gives

$$\frac{p^2+q^2+2pq}{4} - \frac{p^2+q^2-2pq}{4} = pq$$

If *p* and *q* are large prime numbers, they will both be odd so $\frac{p+q}{2}$ and $\frac{p-q}{2}$ are whole numbers.

Suppose $m = x^2 - y^2$ then $x^2 - m = y^2$. This means $x^2 \geq m$ as square numbers cannot be negative so the smallest m could be is $\sqrt{m}$.

We are trying to factorise $m = 655\,427$. $\sqrt{655\,427} = 809.58...$ so the smallest possible value of x is 810.

For each possible value of x find the value of $x^2 - m$. When this is a square number, y^2, the factors are $(x - y)(x + y)$. Doing the calculations on a spreadsheet gives the following:

n	810+n	(810+n)^2-655427=Y	SQRT(Y)	Factor	Factor
1	811	2294	47.8957	no	no
2	812	3917	62.5859	no	no
3	813	5542	74.4446	no	no
4	814	7169	84.6699	no	no
5	815	8798	93.7977	no	no
6	816	10429	102.122	no	no

155	965	275798	525.165	no	no
156	966	277729	527	1493	439
157	967	279662	528.831	no	no

The factors are found after testing 156 numbers whereas the previous method took 439 numbers. It is so quick to do for a 6 digit value of m that it makes little difference but it shows that computing speed and efficient methods of factorising can enable the RSA code to be broken unless the prime numbers used are very large. The current record for successfully factoring a number is 200 digits.

"In the eighties it was generally held that prime numbers of fifty odd digits would suffice. However, developments went much faster than foreseen, and it is a precarious matter to venture upon quantitative forecasts in this field. When Rivest challenged the world in 1977 to factor RSA-129, a 129 digit number (from a special list), he estimated that on the basis of contemporary computational methods and computer systems this would take about 10^{16} years of computing time. Seventeen years later it took only eight months in a world-wide cooperative effort to do the job."

http://www.cwi.nl/en/RSA

Bibliography

Cryptology timeline www.math.cornell.edu/~morris/135/timeline.html

An introduction to cryptography
http://www.math.sunysb.edu/~scott/papers/MSTP/crypto/crypto.html

The Code Book, Simon Singh, Fourth Estate (2000)

Derangements http://www.mathlab.mtu.edu/~eewestlu/ma3210_lecture19.pdf

Codes and ciphers http://www.bletchleypark.org.uk/edu/teachers/ccresources.rhtm

http://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma

Euler function and theorem http://www.cut-the-knot.org/blue/Euler.shtml

Mathematical formulae http://www.po28.dial.pipex.com/maths/formulae.htm

http://www.cwi.nl/en/RSA