An Introduction to Cryptography

Introduction

Secret writing has been used for thousands of years; probably for as long as we have had secrets to keep. These may be messages in war, messages between corporations, or just personal secret messages. Even languages may be thought of as a kind of code.

When we send secret messages there are three types of participants. The *sender*, the intended recipient or *receiver*, and possibly the *interceptor* or 'bad guy'. The act of disguising your message is known as *encryption*, and the original message is called the *plaintext.* The encrypted plaintext is known as the *ciphertext* (or *cryptogram*), and the act of turning the ciphertext back into the original plaintext is *decryption.*

Steganography is the act of physically hiding your message; in ancient times you might write your message on the shell of a boiled egg using special ink that would soak through the porous outer shell. When the egg was cracked and peeled, your secret message would be found written on the egg white. In the 20th century, agents involved in espionage would use the microdot, shrinking their entire message smaller than could be seen by the naked eye.

Codes on the other hand involve turning words or phrases into other words. For example, a command such as 'attack at midnight' might simply become 'eagle'. The problem with codes is that they involve carrying a code book, essentially a dictionary, to translate what you want to say into code. If this book falls into enemy hands then your code is revealed and is now useless, and replacing each code book is not something that can be done frequently. Also, a code may not have the flexibility to deal with all messages you might conceivably wish to send.

In comparison, *ciphers* work on the level of the individual letters of your message. Ciphers may replace letters in a message with other letters, or numbers, or symbols. For example, if 'a' becomes 0, 'b' becomes 1, 'c' becomes 2 and so on, then a word like 'secret' becomes '18 4 2 17 4 19'.

Ciphers comes in two parts: The first part is the *algorithm*; this is simply the method of encryption. The second part is the *key.* The idea is these work together like a lock-and-key and, for the code breaker, having one without the other is just half the problem. The key may change frequently meaning the greater the number of keys the more difficult the cipher becomes break by brute force (an exhaustive check of all possible keys).

Secret writing has always been a constant struggle between the code maker and the code breaker. *Cryptography* is the study of making secret messages, whereas *cryptanalysis* is the study of breaking those secret messages. *Cryptology* is the collective name for both cryptography and cryptanalysis; the word cryptology coming from the Greek words *kryptos* meaning hidden and *logos* meaning word.

1 Monoalphabetic Ciphers

1.1 The Caesar Shift

We begin with a simple cipher as described by Julius Caesar in the Gallic Wars. In it he describes taking two alphabets, from a to z, one above the other. However, the second alphabet is shifted three places to the left. The top alphabet represents letters in the plaintext, while the the alphabet underneath shows what these letters become in the ciphertext.

plaintext	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
ciphertext																										

So, a plaintext message like

veni, vidi, vici

becomes

in the ciphertext.

Clearly, the shift does not have to be 3, (it can be 4, 5, etc). The idea of shifting the alphabet is the algorithm, and the size of the shift is the key. This is not a very secure cipher since there are only keys (including plain English when you shift by). This cipher can be quickly broken by a simple exhaustive check of all possible keys. (The original strength of this cipher may have come from Caesar's enemies not realising he was using a cipher at all).

1.2 General Substitution Ciphers

Remember, a monoalphabetic cipher is a one-to-one rule that pairs each letter in the plaintext alphabet with a letter in the ciphertext alphabet. However, not all monoalphabetic ciphers need to be defined by a formula, like those formulas we have seen for the additive, multiplicative and affine ciphers. Instead, we may define a cipher letter-by-letter, by defining where each letter in the plaintext alphabet is sent in the ciphertext alphabet.

Example. Let's use the first five letters a, b, c, d, e. Then we may define a cipher as;

So a plaintext message such as 'bead' becomes in the ciphertext.

To decrypt this message we simply reverse the direction of the arrows;

We may encrypt a message twice, performing one encryption after another, making a *composite* of ciphers. We can represent this by juxtaposing the two diagrams. We may represent the the result on one diagram by following the total path from left to right for each letter. The result is a new monoalphabetic cipher, however the new cipher is no more secure than the original.

How many monoalphabetic ciphers exist? To calculate this answer consider your choices for each step. Building our cipher from scratch, we have choices for the first letter of the cipher. We then have choices left for the second letter, and choices for the third letter, and so on. Continue this way until we reach the final letter of the cipher, for which we will have one choice left that we have to take.

plaintext	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
choices																										

To find out the total number of keys, i.e. the total number of *combinations* of A to Z, we multiply choices together. Multiplying the integers to inclusive is called *factorial* and is written so

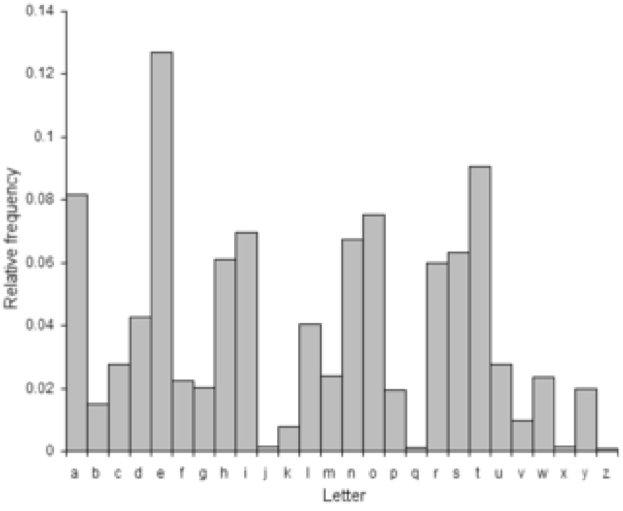
Hence, the total number of key is and is approximately equal to - a very large number indeed. And too many keys to check by brute-force alone.

1.3 Breaking the Cipher: Frequency Analysis

So far we have looked at the roles of the sender and receiver. Now, how will an an interceptor break a monoalphabetic cipher without the key?

Given a long piece of text, one can determine the frequency of each letter. In general, the frequencies of the letters in a given text do not change. Below is the expected frequency of each letter in English:

letter	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
frequency (%)	8.2	1.5	2.8	4.2	12.7	2.2	2.0	6.1	7.0	0.1	0.8	4.0	2.4	6.7	7.5	1.9	0.1	6.0	6.3	9.0	2.8	1.0	2.4	0.1	2.0	0.1



The letters ‘e’, ‘t’ and ‘a’ are the most common, with ‘j’, ‘q’, ‘x’ and ‘z’ the least common. Naturally, frequencies would be different in different languages. Also, frequencies are slightly different depending on the type of message it is, i.e. personal, military or other.

In a monoalphabetic cipher, if the plaintext letter ‘e’ becomes a different letter, say ‘w’, then ‘w’ will now be the most common letter in the ciphertext. The underlying frequencies of the letters remain unchanged, and this is a clue to help us break the cipher. This is called *frequency analysis*.

And there are other tricks we can use: Look for common words like ‘the’ or ‘and’. Look for letters at the ends of words that might be ‘s’. Look for double letters that might be ‘oo’ or ‘tt’; or certain pairs of letters, called *bigrams*, that might be ‘th’ or ‘qu’. Below are the ten most common bigrams and *trigrams*.

1	2	3	4	5	6	7	8	9	10
th	er	on	an	re	he	in	ed	nd	ha

1	2	3	4	5	6	7	8	9	10
the	and	tha	ent	ion	tio	for	nde	has	nce

Example. Let’s use frequency analysis to break:

VCIL NCI VWKIU WDI DSXLT AI HDIWQELB WU E FWYI NCI TIYQ WPSLI
WLT AZ IZI EL KWEL EU UIIQELB USAI BDIIL PIWJ NS DIUN XFSL
VCWN VSXPT LSN E BEKI NS VWLTID VCIDI AZ SPT YSAFWLESLU TVIPP
WHUILYI AWQIU NCI CIWDN BDSV JSLTID EUPI SJ HIWXNZ JWDI NCII VIPP

The ciphertext is 199 letter long, and the individual frequencies are given by:

letter	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
frequency (%)	3	3	4	6	5	2	0	2	18	2	2	8	0	6	0	5	2	0	7	4	5	5	9	2	2	2

The most common letter is `I' which suggests it is the letter `e' in the plaintext. The most common trigram is `NCI', so maybe this is the word `the'. Continuing in this way we will get the final message:

when the waves are round me breaking as i pace the deck alone
and my eye in vain is seeking some green leaf to rest upon
what would not i give to wander where my old companions dwell
absence makes the heart grow fonder isle of beauty fare thee well

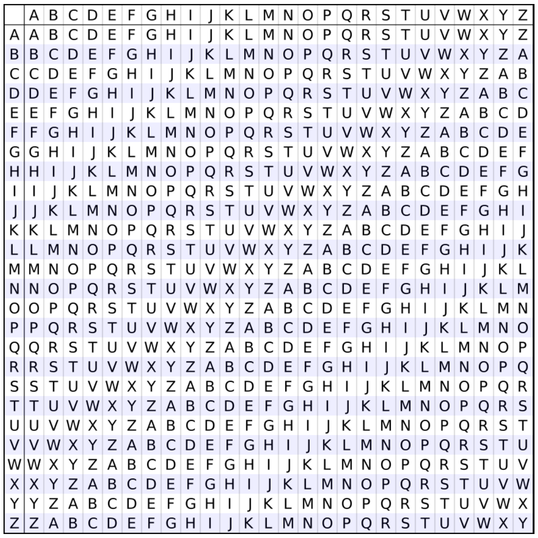
2 Polyalphabetic Ciphers

Monoalphabetic ciphers were used for thousands of years, but were vulnerable to statistical attacks, until a new idea in encryption began to be adopted in order to disguise letter frequencies. A *polyalphabetic cipher* is a rule that uses a different monoalphabetic cipher to encipher a plaintext letter p depending on its position in the plaintext. This means the same letter appearing in two different positions in the plaintext may be enciphered in two different ways, so a double letter in the plaintext may not necessarily be a double letter in the ciphertext.

2.1 Vigenère Cipher

The following cipher is named after the 16th century French diplomat Blaise de Vigenère. Imagine we want to encrypt the message: 'shaken not stirred'.

We start by constructing a Vigenère Square:



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

Notice, each line of the Vigenère Square is a Caesar Shift, starting with a shift of zero, and ending with a shift of 25.

Next we need a *keyword*. Let's use the word . We write that repeatedly above our message:

Keyword:
Plaintext: s h a k e n n o t s t i r r e d
Ciphertext:

To encrypt the first letter: go to row in the square - the first letter of the keyword; next find column - the first letter of the plaintext; and where they meet in the middle is the first letter of the ciphertext . For the next letter do the same again: go to row and column , and where they meet is the second letter of the ciphertext . Continuing in this way we get the ciphertext above.

Notice, the same letter in the plaintext may now be enciphered as different letters in the ciphertext depending on its position. Also notice, two letters that are the same in the ciphertext may come from two different letters in the plaintext.

If the keyword has length , let be the letters of the keyword. If is the plaintext letter in the th position, then the ciphertext letter in the same position, , is given by

Given the keyword, we may decrypt our ciphertext by simply reversing this algorithm.

The effect of the Vigenère cipher is to disguise the frequencies. The longer the keyword the more effect it has on frequency.

2.2 Breaking the Vigenère Cipher

If the keyword used in the Vigenère cipher has length , then every th letter will a result of the same monoalphabetic cipher. If we can determine the length of the keyword then, given a long enough message, we will be able to use standard cryptanalysis tools for monoalphabetic ciphers on every th letter to decrypt the whole message.

If monoalphabetic ciphers are broken by looking for common letters, polyalphabetic ciphers are broken by looking for common words.

Example. Below is ciphertext, 188 letters long, believed to have been enciphered using a Vigenère cipher:

FHVDWSEEMFQCZXVQRZAOBOCGOXPYIPQTZKQUPYMFZADMRMFKMFFHVAWJTVMBFHT
MBFUIGTDEEKVPIGTCYAKJZMIJMRQVZOSZEIMOZDZBKMSVDSZTLIZXYSZCWEEBVD
EVPIZDIMRKERZGXAKMFGSZVUFHVUSFHFLGPEMMZAPVLPKKRAWEKZIBPBRJPMGV

We begin our cryptanalysis by looking for any string of letters that appears more than once in the ciphertext. If we look closely, we can see the string appearing three times in the ciphertext. This is more than a coincidence. Since the keyword is repeated, we proceed by assuming this is a common trigram that just happens to appear under the same three letters of the keyword.

Notice, the distances between the occurrences of is letters, followed by another letters.

The Kasiski Test: If a string of letters appear repeatedly in a polyalphabetic cipher, then the length of the keyword, , may be a common divisor of the distances between occurrences.

So in our example above, the keyword may have length or .

2.5 One Time Pad Cipher

As we have seen, when using the Vigenère cipher, the fewer the repetitions and the longer the keyword the closer the frequencies are to a uniform distribution. The *one time pad cipher* uses a string of random letters, as long as the message itself, as its key.

Example.

Keyword:	**Q**	**F**	**D**	**X**	**S**	**I**	**J**	**L**	**C**	**K**	**S**	**Y**
Plaintext:	a	t	t	a	c	k	o	x	f	o	r	d
Ciphertext:	Q	Y	W	X	U	S	X	I	H	Y	J	B

If you are the interceptor with no knowledge of the key, then it can be proven that this cipher is unbreakable. We will not prove this here, however the idea is that all keys and ciphertexts are equally likely. So, given a piece of ciphertext, and by carefully choosing the key, we could decrypt that ciphertext to make any plaintext message we want. This message could be good, bad, or nonsense - but it doesn't make that message any more likely to be true than it was to begin with.

For example, we may believe there is a 60% chance the message says '*attack oxford*', a 30% chance it says '*attack london*' and a 1% chance the message says '*blah blah blah*' - and there are keys that can decrypt the ciphertext to each one of those messages. However, decryption will not increase the likelihood of any of those messages, to the point that we might as well not bother.

Here the strength of the cipher does not lie in the individual monoalphabetic ciphers (which are additive) but in the randomness of the key. The key must be genuinely random; algorithmic or computer generated random keys will not do. Also, to prevent repetition, each key must be used to encrypt one message only, then discarded.

In reality, the one time pad cipher is impractical to use. When choosing which cipher to use we must balance security with practicality and ease of use. However, the one time pad cipher is used today to send messages between the President of the USA and the President of Russia.

3 Enigma

By the beginning of the 20th century it had become possible, and necessary, to mechanise encryption. In 1918 a German engineer named Arthur Scherbius patented the Enigma Machine. Originally it was sold to banks, railway companies and other organisations who needed to send secret information. By the mid-1920s the German military started to use the Enigma Machine, with some differences from the commercial version of the machine. Enigma was used by the German military throughout World War II, therefore breaking the enigma cipher became a top priority, first by the Polish, then later by the British and Americans.

3.1 Enigma Encryption

The Enigma Machine was an electro-mechanical machine, about the size of a typewriter, made with steel casing inside a wooden box. On the outside you will see two sets of letters which were the keyboard and the lampboard. You would type your message using the keyboard, the message would then be encrypted letter-by-letter, but instead of printing on paper the encrypted letters would light up on lampboard. The encrypted message would then be written down by the operator and would be transmitted by radio. The machine itself did not transmit.

Enigma is a polyalphabetic cipher, where each individual monoalphabetic cipher turns the letters of the alphabet into pairs, also known as a *product of transpositions.* A letter is encrypted as its paired partner.

For example, here are two monoalphabetic ciphers needed to send the message .

input:	a	b	c	d	e	f	g	h	i	j
output 1:	f	i	e	h	c	a	j	d	b	g
output 2:	h	g	j	f	i	d	b	a	e	c

Signatures

The intention of RSA is to encipher using E and decipher using D. But in fact the roles of E and D are interchangeable. That is to say, it is also possible encipher a message with D and decipher a message with E. This has a very important function in authenticating messages.

Consider two correspondents, John and Mary. Each have their own public key, which are known to everyone, and each have their own private key, known only to themselves.

Let's say, Mary's public key is $E = 8023$, $m = 24257$, and John's public key is $E = 8993$, $m = 11413$.

John wants to send the message 'meet you at six john'. Written in blocks this becomes:

me et yo ua ts ix jo hn

or, in numbers:

1204 0419 2414 2000 1918 0823 0914 0713

How can he let Mary know that the message really is from him? One way is for John to use his private key.

John enciphers his signature using his private key so it reads **2981 6024**:

1204 0419 2414 2000 1918 0823 **2981 6024**

John now enciphers the message completely using Mary's publicly key:

5141 12103 10542 3905 15868 21143 **0124 15283**

On receiving the message, Mary decrypts it using her private key to produce:

1204 0419 2414 2000 1918 0823 **2981 6024**

Mary extracts the signature **2981 6024**. Knowing that the message is supposed to be from John, Mary applies John's public key to his signature to get **0914 0713**.

Finally, putting all this together, Mary gets the message: 'meet you at six **john**'. And the only one who could have made that possible is John himself.

Note, it is more secure to use different RSA keys for encryption and signing, and padding should be used for message signing as it is for message encryption.

The Future

As we have seen, public key cryptography allows two people, who might never have met before, to communicate securely. This is done by publishing a public key, and keeping a private key that cannot be determined from the public key. In the case of RSA, this is due to how hard it is to factorise large numbers.

The two fastest known algorithms for factorising numbers are the *quadratic sieve* and the *general number field sieve.* Current RSA keys are 2048-bits (i.e. 617 digits) long, which can take years, or even thousands of years, to factorise. However, future advances in *quantum computing* may change that.

Traditional computers store information as 1s and 0s. However, quantum computers store information as 1s and 0s simultaneously. This allows quantum computers to perform algorithms that exploit the probabilistic rules of quantum physics. One such algorithm is *Shor's algorithm* that can efficiently determine prime factors, thus breaking RSA encryption.

Other forms of public key encryption, such as *Diffie-Hellman key exchange* and *elliptic curve cryptography*, are also quantum-breakable. So the future of cryptography will depend on devising problems that are provably hard for quantum computers.

One candidate for a quantum-secure system is *lattice-based cryptography.* Security here is related to the difficulty of finding the nearest point in a lattice with hundred of spatial dimensions (associated with the private key) given an arbitrary location in space (associated with the public key).

Other potentially quantum-secure systems include *code-based cryptography* and *multivariate cryptography.* Both rely on problems that are (presumably) difficult for quantum computers.

Quantum computers are still in their early days, and do not yet exist in any practical form. However, there are other technologies, available today, that do use the properties of quantum physics for sending secret information. These methods are collectively known as *quantum cryptography.*

The most well-known example of quantum cryptography is *quantum key exchange.* Here, the key is transmitted by particles of light through fibre optic cables. However, according to quantum physics, measuring particles of light changes them. So, a portion of the key is compared before and after transmission. Then, if someone has attempted to spy on the transmission, this act can be detected and the key is not used.

Once the key has been established, the message is encrypted using classical techniques. For example, the encryption method could be a one-time pad, which would make decryption impossible. Practical problems with quantum key exchange include transmission distance and key generation rate limitations.

Another example of quantum cryptography, and one that is purely quantum-based, is *Kak's three-stage protocol*, a form of three-pass protocol that uses particles of light to encrypt and transmit a message directly.

Cryptology used to be a problem for linguists. Later it became a problem for mathematicians and computer scientists, and now optical engineers. Who will be the next code breakers?