Modern Cryptography and Mathematics

By Michael Manfredi a third year student at The University of Warwick

ACCESS TO FURTHER mathematics
the **Further Mathematics** network
www.fmnetwork.org.uk

What is Cryptography?

According to the Oxford English Dictionary cryptography is, "A secret manner of writing ... intelligible only to those possessing the key; also anything written in this way. Generally, the art of writing or solving ciphers." A secret code used in cryptography is called a cipher, the process of using a cipher to turn a plain document into a secret text is called encryption, and the reverse process is called decryption.

Cryptography is an ancient subject that has changed a lot throughout the years. To quote Wikipedia: "In modern times, cryptography is considered to be a branch of both mathematics and computer science." At one time the subject was mainly a linguistic one, the key concern being the ability to recognise words and make words unrecognisable with a simple cipher. Today, thanks to computers, the subject is very mathematical with the ciphers involved drawing from computer science and number theory.

Problems with communication

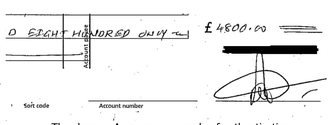
There are several issues that can arise when communicating over vast distances. The methods of modern encryption aim to solve two of them:

Confidentiality:

Will any information sent only be read by it's intended recipient? The world, both past and present, is full of confidential information, from bank account details to military strategies. There is always the risk that such information could fall into the wrong hands. Confidentiality can be achieved without cryptography, by using a reliable courier for example, but the modern internet is anything but secure with any information sent on it usually being routed through various computers around the world before reaching its destination.

Authentication:

Is any information received from the person it claims to be from? A naval captain receiving an order to scuttle his ship might wonder if the order came from his enemy. A bank confronted with a cheque might wonder who it was written by. In the context of cheques and paper documents and in the context of modern cryptographic techniques the authentication is usually called a "signature" or sometimes a "digital signature". With a modern desire to communicate using the internet, authentication becomes a major issue.



The cheque: A common example of authentication.

History: Simple Substitution

Simple substitution, as the name implies, involves substituting some letters for others, as in this table:

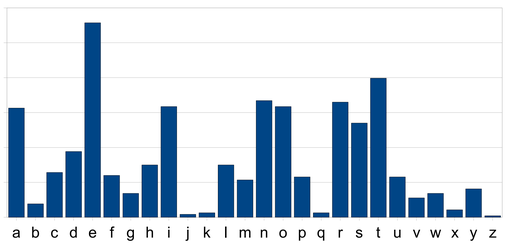
Substitute every letter in the top row with one in the bottom:

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

"this text is going to be encrypted" becomes "jdcu jsoj cu ebcge jb as sgznlyjsw"

Historically, once both parties have agreed upon a substitution, this method offered a reasonable degree of confidentiality. To anyone intercepting the message it would appear to be nonsensical. It also offers a degree of authentication since (hopefully!) only the two parties who wish to communicate know which substitution to use.

While likely to be able to fool the primitive civilisations of the past, this method is insecure by modern standards. By counting which letters are most common it's easy to make an educated guess which letters stand for which. The graph below shows the relative frequency of each letter in English for a typical piece of text. This pattern is a great weakness:



Introduction of computers

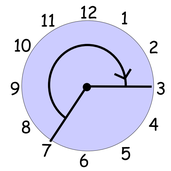
The introduction of computers has changed the face of encryption. Historically encryption was done by hand with pencil (or pen) and paper. A typical modern personal computer will be able to perform around 5 billion "operations" per second. This processing power coupled with its widespread use has changed the face of encryption. Not only is this processing power a boon to encryption, opening the doors to otherwise long and tedious mathematical calculation, it is also a threat, providing a resource an eavesdropper could utilise to attempt to break an encryption.

Fun Mathematics: Modular Arithmetic

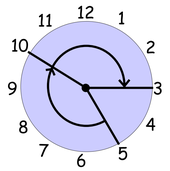
Modular Arithmetic, sometimes called clock arithmetic is part of a branch of mathematics called number theory. This branch of mathematics has found many applications in cryptography.

As an example: if it is 7 o'clock what time will it be 8 hours from now? $7 + 8 = 15$, but since we're working with a clock face we have to reduce 15 down to a proper time by subtracting 12, to get 3 o'clock i.e. 3 is the remainder when 15 is divided by 12. Mathematicians would write this as:

$$7 + 8 \equiv 15 \equiv 3 \pmod{12}$$



$7 + 8 \equiv 3 \pmod{12}$



$5 + 5 + 5 \equiv 3 \pmod{12}$
$3 \times 5 \equiv 3 \pmod{12}$

Multiplication is similar to addition: $3 \times 5 \equiv 5 + 5 + 5 \equiv 15 \equiv 3 \pmod{12}$
Subtraction is addition backwards: $7 - 8 \equiv -1 \equiv 11 \pmod{12}$
We can define powers in the natural way: $3^4 \equiv 3 \times 3 \times 3 \times 3 \equiv 81 \equiv 9 \pmod{12}$

Division is far more tricky:
Division is thought of as the "inverse" to multiplication. For example if

$$5a \equiv 1 \pmod{12}$$

then multiplication by a is the same as division by 5. Ignoring modular arithmetic one would say

$$a = 1/5 = 0.2$$

but 0.2 isn't on the clock face, it's not even a whole number. If we let $a = 5$ we notice that

$$5 \times 5 \equiv 25 \equiv 1 \pmod{12}$$

which solves our problem. Naively one could say: " $1/5 \equiv 5 \pmod{12}$ "

Division has other problems. In clock arithmetic:
$8 \times 2 \equiv 4 \pmod{12}$ but $8 \times 5 \equiv 4 \pmod{12}$
Given the problem: $8a \equiv 4 \pmod{12}$ there is no way to divide through by 8 and recover a. a could be equal to 2 or 5 (or 8 or 11). One could say 1/8 does not exist, a bit like 1/0 in the usual number system.

There's no reason why we use 12, the same can work for any number, n. In the general case we call the collection of numbers and the arithmetic that goes with them "the integers modulo n". These are a very powerful tool in number theory, and a lot is known about them. It turns out that if we choose n to be a prime number then the problems with division disappear. Other nice things can happen, for example:

Fermat's little theorem:

If p is a prime number then

$$a^p \equiv a \pmod{p}$$

Removing the language of modular arithmetic; this is equivalent to saying that if p is prime then p is always a factor of $(a^p - a)$. Another example:

Chinese remainder theorem:

If n and m have no common factor then there is a solution x to:

$$x \equiv a \pmod{n}$$
$$x \equiv b \pmod{m}$$

For example; if my age, when divided by 3 gives remainder 1, and when divided by 5 gives remainder 2, then my age is a solution to:

$$x \equiv 1 \pmod{3}$$
$$x \equiv 2 \pmod{5}$$

7 is a solution, and so is 22, 37, 52, 67, 82 ... etc

Encryption: Symmetric Encryption

The encryption/decryption process is split into four parts:

- The encryption algorithm
- The encryption key
- The decryption algorithm
- The decryption key

In the historical example we had the encryption algorithm:
"Replace every letter appearing on the top row of the key with the letter below it".

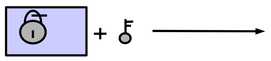
And the encryption key:

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

The decryption algorithm was just the encryption one in reverse:
"Replace every letter appearing on the bottom row of the key with the letter above it".
The decryption process is similar; it has an algorithm and a decryption key. Usually they are very similar, and if you know the encryption process it is trivial to work out the decryption process. If it's trivial to work out the decryption key from knowing the encryption key then the encryption is said to be symmetric. As the name symmetric suggests encryption and decryption mirror each other.

Most encryption is like this, and it makes logical sense that once you know how a message is "jumbled up" then the "un-jumbling" process is just the same but in reverse and fairly easy to work out. As in the historical example if your encryption process replaces every letter A with the letter Q, then the decryption process replaces every letter Q with the letter A. This form of encryption has a big disadvantage. If you want to send an encrypted private message to someone far away, you have to first tell the other person exactly how to decrypt it and the encryption would be useless should the message detailing how to decrypt it be intercepted.

This is analogous to saying that if you want to send someone a locked box then you must also send them the key to it, for without it they cannot open it, but should both the key and the box be intercepted the whole locking process is useless.



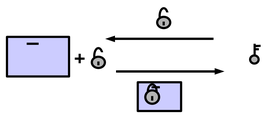
For symmetric encryption, authentication and confidentiality are the same thing; only the sender and receiver will know the key, so any message the receiver receives must have come from the sender, because only the sender knows the key.

Public Key Cryptography

Public key cryptography is a modern technique. The encryption and decryption processes have different keys and if someone knows the encryption key it's very, very difficult for them to work out the decryption key. The word asymmetric is sometimes used to describe this type of encryption to highlight the fact that encryption and decryption do not mirror each other, their keys are distinct and knowing the encryption one doesn't allow you to (easily) work out the decryption one.

In public key cryptography the encryption key is made public and called the public key. The decryption key is made private and called the private key. Anyone can get hold of a copy of the public key and use it to encrypt a message that only the intended recipient, the only person with the private key, can decrypt.

This is analogous to a similar system with lock boxes, padlocks and keys. The public key is like an open padlock. Someone wanting to receive a locked box can send out an unclasped padlock. The person wanting to send the locked box can use this padlock to lock it and send the locked box back to the sender, who has the key.



Digital Signatures

One newer idea in the area of cryptography is that of digital signatures. With public key cryptography the argument:

"I know this message came from Mark, because only he knows the secret code".

Is invalid because by the nature of public key cryptography the encryption key is public therefore everyone knows the encryption key! This means authentication is a real issue, and a very separate one from confidentiality.

The way around this is digital signatures, which are basically the reverse of public key cryptography. The encryption key is private, but the decryption key is made public. This means only one person, call him Mark, knows how to encrypt a message but everyone else can decrypt it. Any message encrypted using Mark's key must have come from Mark because nobody else knows the key. This is of course assuming it's impossible to work out Mark's encryption key from knowing the public decryption key.

Diffie-Hellman Key Exchange

After a collaboration between Ralph Merkle, Whitfield Diffie and Martin Hellman in the 1970s, Diffie and Hellman published a paper describing a method two people could use to agree upon a key for use in one of the many symmetric encryption methods in existence. This method was called Diffie-Hellman key exchange.

One person, call her Jane, can contact another person, call him Mark. Then, using this key exchange method, after a dialogue they can both have agreed upon a key to use with a symmetric encryption system. Any eavesdropper listening to this dialogue would be unable to work out what key the two agreed upon!

This was a breakthrough in cryptography that solved an ancient Achilles' heel of encryption; the difficulty of two people agreeing upon an encryption key without an eavesdropper finding out what it is.

How it works:

If Jane and Mark wish to use Diffie-Hellman, they agree upon a large prime number, call this number p, and a number between 1 and p, call this number g. We know for some number n that:

$$g^n \equiv 1 \pmod{p}$$

For security we want the smallest n for which this is true to be very large. The larger the better.

Armed with the numbers p and g Mark and Jane can continue as follows:

Jane picks a number, a, and computes $g^a \pmod{p}$ and sends this number to Mark.

Mark picks a number, b, and computes $g^b \pmod{p}$ and sends this number to Jane.

Jane then knows what $g^b \pmod{p}$ is, and can raise this to the power of a to get $(g^b)^a \pmod{p}$

Mark then knows what $g^a \pmod{p}$ is, and can raise this to the power of b to get $(g^a)^b \pmod{p}$

$$(g^a)^b \equiv g^{ab} \equiv g^{ba} \equiv (g^b)^a \pmod{p}$$

Therefore both Mark and Jane know the same number: $g^{ab} \pmod{p}$, which they can use as a key for any encryption method.

The only bits of information sent back and forth between Mark and Jane are the values of:

p, g, $g^a \pmod{p}$, and $g^b \pmod{p}$

From this information it is computationally very, very difficult for an eavesdropper to recover the value of $g^{ab} \pmod{p}$, which Mark and Jane know.

One problem that arises when using Diffie-Hellman is that of authentication. Jane may think she is doing the key exchange with Mark, when in fact she is communicating with someone pretending to be Mark.

RSA Algorithm

The RSA algorithm is an algorithm developed by Ron Rivest, Adi Shamir, and Leonard Adleman.

The algorithm was first publicly described in August 1977 and a patent was filed in December 1977; as the patent was filed after the discovery was made public, the algorithm received a patent in the US only in 1983.

Above: Adi Shamir, Ronald Rivest, and Leonard Adleman.

The RSA algorithm is the first example of an algorithm for public key cryptography. It also solves the problem of authentication with public key systems.

Much like Diffie-Hellman, RSA draws on modular arithmetic. It's security mainly relies upon the difficulty of factorising a large number which is a product of two large primes. The two primes are part of the private key, whereas the single large number, the product of these two smaller ones is part of the public key.

Multiplying two prime numbers together is computationally like a one way process. It is easy to calculate:

$$757 \times 977 = ?$$

But given the product it's comparatively very difficult to factorise it to recover the two primes:

$$739\,589 = ? \times ?$$

In practice the primes used by RSA are several hundred digits long, so methods to find such large primes are needed; fortunately such methods exist. Since the security relies on it being difficult to factorise large numbers we are fortunate (or unfortunate) there is no method to quickly factorise numbers.

A company called "RSA, The Security Division of EMC" offers large rewards to anyone capable of factorising a collection of large numbers. The latest of these numbers to be factorised was the 193 digit number RSA-640 in 2005 for a reward of $20,000. RSA-704 (212 digits) and RSA-768 (232 digits) are still open and carry rewards of $30,000 and $50,000 respectively.

Financial transactions over the internet are completely dependent on public key cryptography.

Produced by the Further Mathematics Network, **www.fmnetwork.org.uk** and The University of Warwick, **www2.warwick.ac.uk/fac/sci/maths** with the support of Rolls-Royce plc, **www.rolls-royce.com**