**Encryption in Transmitting Information**

Encryption is re-arranging data such that it can only be understood by the intended recipient/s.

There are two ways of doing this – codes and cyphers. Codes replace commonly understood ways of transmitting information with something only the recipient knows about. For example, Morse code uses dots and dashes instead of letters. Simply translating something to another language could be counted as code. In fact, this was done during WWII – sensitive messages were transmitted in Navajo, a language only a handful of people in the world knew.  
The problem with codes is that if you know roughly what the message says, it is easy to spot patterns and figure out how it works.

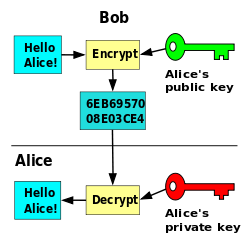
Cyphers re-arranges the contents of the message instead. The simplest cypher dates back to roman times, and is called the Ceaser Cypher, as it was used by a roman emperor called Ceaser.

It simply involves transposing every letter in the message up or down by a fixed number. For example, if transposed by +1, ‘cat’ becomes ‘dbu’. When transposed by -3, ‘dog’ becomes ‘ame’.

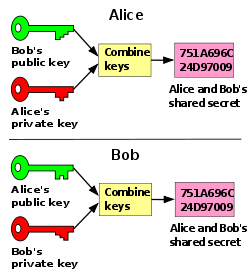
There are only 26 possible Ceaser cyphers (one of which is the original anyway), so it is not particularly secure.

Modern cyphers employed by computers use much more sophisticated maths to encrypt data.

The most common is probably Public Key Encryption, or Asymmetric Encryption. The standard implementation of this encryption uses two randomly generated prime numbers – these are the ‘private key’. These are then multiplied to create a public key. There is currently no known method of reversing this multiplication process (working out which two primes were multiplied). When somebody wants to send some data, they request the public key of the recipient. They then encrypt the data using the public key and send it. The recipient can then decrypt the data using their private key. Anybody who intercepts the data will not be able to use it, as they do not have the private key.



There are variations on Asymmetric Encryption. For example, In the Diffie–Hellman key exchange scheme, the two parties swap public keys, then combine the partners public key with their own private key, creating a ‘shared secret’. Both parties can compute the secret independently, as they have the partners public key. The information is both encrypted and decrypted using this secret.



This is the asymmetric version of symmetric encryption. In symmetric encryption, only one key is used for both encrypting and decrypting data. Both parties must have this data before secure communication can take place.  
This causes the ‘locked box’ problem – how can you securely send the key between parties without it being compromised if it is intercepted?  
The real-world analogy for this is how to send a locked box to someone but enable them to open it. You could send the key, but that must also be locked in a box to secure it, and so on ad infinitum.  
The solution is to send the locked box (encrypted message). The recipient then adds their own lock to the box, and sends it back to you. You then remove your lock, and send it to the recipient, who can unlock the box with their own key (decrypt the message).

Encryption can be used in multiple ways. In the above examples, it was about transmitting information.  
This would often be done when sending any kind of sensitive information over a network or the internet. The HTTPS protocol allows you to send login credentials without them being intercepted, for example.  
However, data that is ‘at rest’ (i.e not in transit or use, such as data on a hard drive or memory stick) can be encrypted too. In fact, the Windows OS comes with a tool called BitLocker specifically for this purpose. It can encrypt whole drives, partitions or individual files and folders.  
This data can be decrypted and used on the fly, making it a secure way to use sensitive or valuable data.