**CODES**

What does Julius Ceasar have in common with the transmission of modern digital signals? The short answer is codes and coding.

Accuracy was very important for Ceasar and it is also required for the effective transmission of digital signals. Ceasar also wanted to keep his codes to himself as do the cable and satellite broadcasting television companies who only want paying subscribers to be able to watch their program.

Error detection and correction

Let's look accuracy first. Human error or „noise along the line“ can always ocure during the transmission of messages and must be dealth with. Mathematical thinking allows us to construct coding systems that automatically detect errors and even make corrections.

One of the first binary coding systems was the Morse code invented in 19th century by Samuel Morse and it was designed and used for sending messages over the telegraph. It consists of two symbols: dash and dot. But Morse code isn't good in error detection and correction. If we wanted to send „CAB“, but mistyped a dot for a dash in C, forgot the dash in A and noise in the wire substituted a dash for dot in B, the receiver would get „FEZ“ and wouldn't notice the error.

At more primitive level we could look at coding system consisting of just 0 and 1, where 0 represents one word and 1 another. Suppose an army commander has to transmit message to his troups, where 0 means „do not invade“ and 1 means „invade“. If a 1 or a 0 was incorrectly transmitted the reciever would never know and the wrong instruction would be given with disadtrous consequences.

We can improve matters by using code words of length two. If this time we code the „invade“ instruction by 11 and „do not invade“ by 00, this is better. The advantage of this system is that an error would be detectable, but we still wouldn't know how to correct it. If 01 or 10 were received we would know that it's incorrect because only 00 and 11 are legitimate words, but we wouldn't know does 01 represent 11 or 00.

The way to a better system is to combine design with longer code words. If we code the „invade“ instruction by 111 and „do not invade“ by 000 an error in one digit could certainly be detected, as before. If we knew that at most one error could be made, the correction could actually be made by the reciever. E.g. if 110 were received then the correct message would have been 111. In this system there are only two code words but they are far enough to make error detection and correction possible.

The same principle is used when word processing is in autocorrect mode. If we type „animul“ the word processor detects the error and corrects it by taking the nearest word, „animal“. But if we type „lopm“ there is no unique nearest word; the words lamp, limp, lump are all equidistant in terms of single errors from lomp and it can't be autocorrected.

Making messages secret

Julius Ceasar kept his messages secret by changing around the letters of his message according to a key that only he and his generals knew. If the key fell into the wrong hands his messages could be deciphered by his enemies. In medieval times, Mary Queen of Scots sent secret messages in code from her prison cell. Mary had in mind the overthrow of her cousin, Queen Elizabeth, but her coded messages were intercepted. Her codes were based on substitutions but ones whose key could be uncovered by analysing the frequency of letters and symbols used. During the Second World War the German Enigma code was cracked by the discovery of its key. In this case it was a very hard challenge but the code was always vulnerable because the key was transmitted as a part of the message.

Public key encryption

An astonishing development in encryption of messages was discovered in the 1970s. Running counter to everything that had been previously believed , it said that the secret key could be broadcasted to all and yet message could remain entirely safe. This is called public key cryptography. The method depends on a 200 year old theorem in a branch of mathematics glorified for being the most useless of all.

E.g. Mr John Sender, a secret agent known as „J“, has just arrived in town and wants to send Dr Receiver a secret message to announce his arrival. He goes to public library takes the town directory off the shelf and looks up Dr Reciever. In the directory he finds two numbers alongside Receiver's name – a long one, which is 247, and a short one 5. This information is available to all and it's the only information Sender needs to encrypt his message, which is his calling card „J“. This letter is number 74 in list of words, again publicly available. Sender encrypts 74 by calculating 745(modulo 247), that is, he wants to know reminder on dividing 745 by 247. When he divides those numbers he gets the remainder 120. Sender's encrypted message is 120 and he sends it to Reciever. Because the numbers 247 and 5 were publicly available anyone could encrypt the message, but not everyone could decrypt it. Dr Reciever has more information. He made up his personal number 247 by multiplying p=13 and q=19, but he is the only one who knows it. This is where the ancient theorem due to Leonhard Euler comes up. Dr Reciever has the knowledge of p=13 and q=19 to find a value of a where 5 x a ≡ 1 modulo (p-1)(q-1) where the ≡ symbol means equals in modular arithmetics. What is a so that dividing 5 x a by 12\*18 gives reminder 1? The answer is a=173. Now, he works out the reminder when he divides the huge number 120173 by 247. The answer is 74 as Euler knew 200 years ago. Reciever looks up word 74 and sees that J is back in town.

You might say, surely a hacker could discover the fact that 247 = 13\*19 and the code would be cracked. And you would be correct. But the public key system is secure and if the might of supercomputers joined together are sucessful in factoring an encryption number, all Dr Reciever needs to do is increase its size still further. In the end it is much easier for Dr Reciever to „mix boxes of black sand and white sand together“ than for any hacker to unmix them.