

Keeping secrets secret



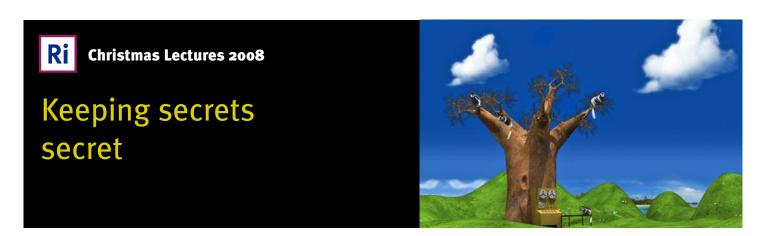
One of the most important concerns with using modern technology is how to keep your secrets secret. For instance, you wouldn't want anyone to intercept your emails and read them or to listen to your mobile phone conversations. This is especially important when someone is sending their credit card details to an online shop. And it is crucial that banks, big businesses and governments know that any information they send through the internet stays safe.

You can protect information by *encrypting* it using a special *key*. This means that the text in your email is rewritten in such a way that no one else can read it, unless they know the key that was used. Throughout history, people have used different ways to encrypt information, and encryptions have become more complex as people have 'cracked' a particular code. Nowadays most of the sensitive information sent via the internet is encrypted using a key that is 128 bits long – a number as large as a 1 followed by 38 zeros!

The process of protecting information is called *cryptography*, and here we explore cryptography in two ways. First, we'll look at hidden pictures that only reveal themselves when you pair them up with their correct partner, or *visual cryptography*. Then we'll take a look at how information on the internet is encrypted using *clock arithmetic*, a kind of maths that is really easy to do in one direction but almost impossible in reverse.





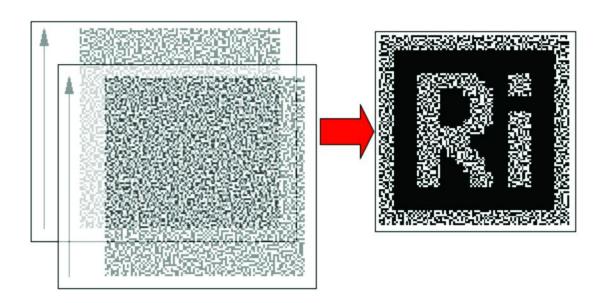


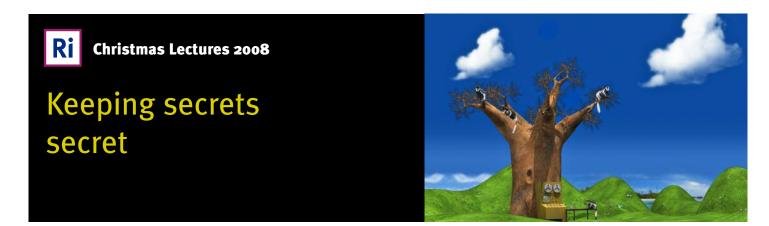
VISUAL CRYPTOGRAPHY

Visual cryptography, first described by two mathematicians in 1994, can be used to encrypt any form of black and white image – printed text, handwritten notes and photos. The best bit is that you don't need to use any difficult maths to decode the hidden picture – you just line up two sheets of acetate and the image appears before your eyes!

Any image can be made up of a grid of picture elements, or pixels (see 'In the Picture'). Visual cryptography splits the pixels that make up your picture into two sets: one called the cryptosheet and the other called the keysheet. These two sheets are printed on acetate so that the black pixels show through the gaps when you lay them one on top of the other. Each sheet shows a random spread of black and white dots but when you match the acetates of the cryptosheet with the correct keysheet, the pixels line up perfectly to reveal the hidden picture.

Lining up the cryptosheet and the keysheet is easier when they both have an arrow on the side. These let you know which way round the sheets should be. The diagram shows how to align the arrows on a cryptosheet with its correct keysheet to reveal the hidden picture (here, the RI logo).





Producing your own visual cryptography is easy with the help of a computer, and the keysheet is made first. For each individual pixel in the secret image (either a black or white square) a 2 x 2 mini-grid is created on the keysheet. Each mini-grid is randomly set to one of six patterns, which show as a horizontal, vertical or diagonal black line, as shown below.



Next you make the cryptosheet. Like the keysheet, each pixel in your image is split into a 2 x 2 mini-grid. But this time you don't set the pattern on each mini-grid randomly, instead you choose them so that the mini-grid pattern on the cryptosheet is exactly the same as or complementary to that on the keysheet:

- where the two mini-grids are the same, the black and white squares will overlap perfectly and produce a mini-grid that is half-white
- where the two mini-grids are complementary (like the two horizontal line patterns that are shown first and second above), the black squares combine and show a completely black mini-grid.

By carefully choosing which mini-grids to use on the cryptosheet you can reproduce the black and white pixels (the white pixels are actually grey – mini-grids half black and half white) of your secret image when the cryptosheet and the random keysheet are overlapped.

Neither sheet contains any information about the hidden image – they just look like random black and white dots. It is only when you match the correct cryptosheet with its keysheet that the black pixels line up to reveal the secret picture.





So, if you wanted to encrypt some information such as a pirate treasure map, you would use visual cryptography to split the map into two acetate sheets. Give one acetate to a friend (it doesn't matter whether it's the keysheet or the cryptosheet) and keep one yourself. Neither of you can reveal the treasure map without using the other person's sheet. And, should another pirate capture either of the two sheets, they could never discover where your treasure is!

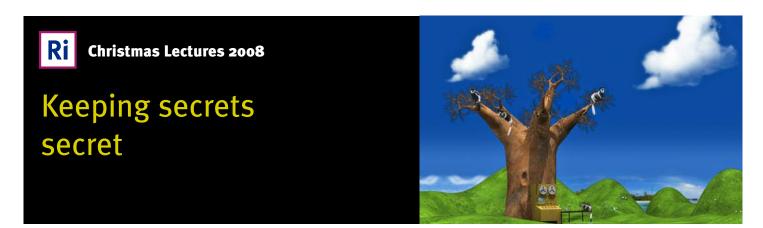
In fact, because you choose which mini-grids to use for your secret image when you are making the cryptosheet you can use the same random keysheet to reveal the hidden pictures on lots of different cryptosheets. Ask your parent or a teacher to print out pages 8 and 9 onto acetate. The first is the keysheet and the second contains two separate cryptosheets – each with a different pirate treasure map that can be revealed using the same keysheet!

Don't worry if you don't understand everything about how the keysheet and cryptosheet are made – you can still have loads of fun with the visual cryptography game.

Visual cryptography game

First, you'll need to ask your teacher or parent to print the visual cryptography game sheets below onto acetate. Each of the pages has two different grids so cut them in half so that there are enough for everybody.

Second, mix up the grids and give one to each person. Challenge them to find their cryptographic partner as quickly as possible by lining up their acetate sheets. Explain that the secret image will appear only when their cryptosheet is paired with the correct keysheet. They must test different partners to find the perfect match. The two people who are first to find each other are the winners; the last are the losers.

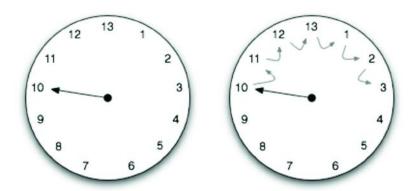


CLOCK ARITHMETIC

Visual cryptography is a lot of fun, but it's no good for encrypting information to send via the internet, for example. Instead, we need a way to change our message into a secret code that is easy for us to do, but hard for someone else to undo (unless they know our secret key). One way to encrypt information is to use *modular arithmetic*, which has a lot to do with clocks and telling the time!

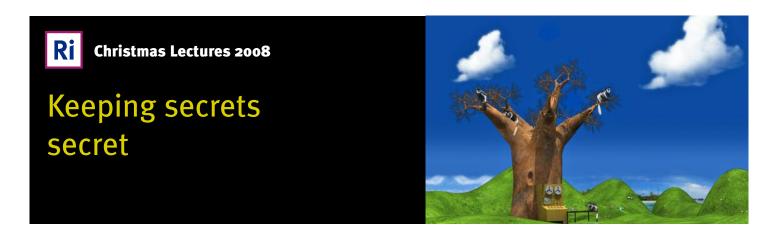
Modular arithmetic is used for counting, but the numbers wrap around once they reach a certain value. We use this kind of arithmetic whenever we tell the time. For example, if I ask you what the time is 3 hours after 10 am, you'd say 1 pm. The hour hand passes 12 and gets back to 1 again. Here 12 is called the *modulus* because it is the number that wraps around. That's why modular arithmetic is nicknamed 'clock arithmetic'.

This means we can use clock arithmetic for encrypting information, but we'll use a clock face that includes the number 13. Using the picture below, if we start at 10 and add 6 we get to number 3. So using this clock arithmetic, we would say: 10 + 6 = 3 (not 16).



Counting upwards using this clock arithmetic is easy – you just remember to go back to 1 when you reach 13. Other maths operations in clock arithmetic (such as multiplication) are similar to standard arithmetic. If you want to double a number in clock arithmetic all you do is count onwards that number of places.

For example, if we want to double 10 in clock arithmetic, using the above clock: start from 10 then count on another 10 places. You should get to the number 7. So, in clock arithmetic: $10 \times 2 = 7$.



Using this clock arithmetic, if we start from 1 and double it each time, we get this sequence of numbers:

Step	1	2	3	4	5	6	7	8	9	10	11	12	13
Result	1	2	4	8	3	6	12	11	9	5	10	7	1

The result row tells you what number you reach after starting with 1 and doubling the number of times shown in the 'Step' row. With this clock face, after 12 steps you'll notice you've visited every number before getting back to 1 and starting again. This doesn't happen with a clock face that has modulus 12; and is because our special clock face has a prime number -13 – as its top number.

The important thing about doubling up in modulus 13 is that the order you visit the numbers is hard to predict. For example, I start from 1 and double up a secret number of times and the answer comes out as 11. You need to work out the secret number of doublings that I used. This is the same as me telling you the 'result' is 11 and you have to work out the 'step' number, as in the table above.

In standard arithmetic this would be easy. If I fold a sheet of paper in half once I get two layers; fold it over again I get 4, then 8, then 16, 32, and so on. So, asking how many doublings result in 32 is the same as asking how many times have I folded a piece of paper that ends up 32 layers thick. In standard arithmetic calculating the number of layers produced by doublings is easy. As is going backwards and working out the number of doublings needed to give a certain number of layers. But in clock arithmetic working backwards to find the number of doublings needed to give a particular result is really hard, because the numbers loop round again. So if I told you the result was 11, you could work out the secret number of doublings by looking all the way along the result row of your table and seeing which step number gives 11. There is no shortcut – you always have to find the answer by working out the whole doublings table and then using it to find the correct step number.

So, the point about clock arithmetic is that it's really easy for me to do a few doublings and tell you the result, but it's really hard for you to work backwards to find what my number of doublings was. Doubling in clock arithmetic is like a 'one-way street' – it's easy one way but difficult going the other – which is why it's useful for encrypting secret information.



Keeping secrets secret



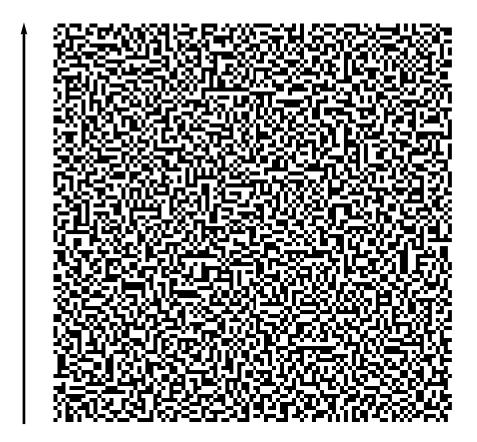
Let's say I want to encrypt my bank card PIN to protect it: I'll use the one-way nature of clock arithmetic. Working out the number of doublings needed to give a particular result is difficult for someone trying to crack it, so I'll take each digit of my PIN and double it that many times. The answer comes out as 3-9-12-11.

Can you crack the code and work out what my PIN is?

To do so, you must look all the way through the doubling table above to find each of these results. Even though I've used a one-way function like clock arithmetic to encrypt my PIN, it doesn't take you all that long to crack the code. Now imagine that I use a special clock with a much larger top number than 13 (but which must still be a prime number), and I multiply by a bigger number each time. If you want to crack the new code to my PIN, it'll take you ages to work out the whole results table just to look up the answers.

In fact, the kind of encryption commonly used across the internet is designed so that without knowing the key, even the fastest supercomputers in the world would take thousands of years to work it out!

PIRATE TREASURE MAP: KEYSHEET



PIRATE TREASURE MAP: CRYPTOSHEETS – X MARKS THE SPOT!

