STEM LEARNING

Quantum Technology
PROGRAMME

ACTIVITY SHEETS

Quantum Encryption Keys

SHARING DECRYPTION KEYS AND DECODING MESSAGES

Every time you send an email or you pay for something online, or with your debit card, a secure communication channel needs to be established between the sender and the receiver, so that nobody else can see the information being shared and steal it.

The first step is to share the decryption key. But in order to do this the sender needs to be sure it can trust the receiver. So, the receiver needs to signal the sender that they can be trusted.

In this activity your teacher will act as the sender and you will be the receiver and we will simulate this first step with a simple 'shift cipher' (or 'Caesar cipher'). If you can decode the message below, your teacher will send you the key to decode the second message. In this first stage a virtual handshake is established by showing to the sender that the person asking for the key is the intended recipient. In RSA public key distribution things are much more complex than in our example, but the analogy is helpful to understand this process.

The encoded message is: VZFSYZR UMDXNHX NX FBJXTRJ

Decode the message and tell your teacher to obtain the decryption key for the message below.

UK NATIONAL QUANTUM TECHNOLOGIES PROGRAMME

HOW DO WE USE ENCRYPTION KEYS?

If you decoded the previous message your teacher will have given the decryption key for the next activity.

The decryption key is:

Follow the rules below to convert your key into a binary key:

1. Every vowel becomes a 1
2. Every consonant becomes a 0

So, the binary key is:

Now apply a XOR operation for each character in the encoded binary message below to decode the message by looking up the binary letters in the ASCII grid provided.

The XOR operation can be summarised as:

0 0 = 0
0 1 = 1
1 0 = 1
1 1 = 0

So, you will first need to convert each binary 7-digit code into the correct 7-digit binary letter in the encoded message below, then convert these into real letters using the ASCII conversion table.

Encoded binary message:

11000010 11000111 11010010 11011101 11000111 11000111 11011110

11010000 11010110 11001010

11010111 11011010 11000000 11000111 11000001 11011010 11010001 11000111 11000111 11011010 11011100 11011101

Decoded binary message:

_ _

_ _

_ _

_ _

Decoded message in English:

_ _ _ _ _ _ _ _ _ _ _ _ _ _

_ _ _ _ _ _

_ _

ASCII conversion table:

Letter	ASCII code	Binary
A	065	01000001
B	066	01000010
C	067	01000011
D	068	01000100
E	069	01000101
F	070	01000110
G	071	01000111
H	072	01001000
I	073	01001001
J	074	01001010
K	075	01001011
L	076	01001100
M	077	01001101
N	078	01001110
O	079	01001111

Letter	ASCII code	Binary
P	080	01010000
Q	081	01010001
R	082	01010010
S	083	01010011
T	084	01010100
U	085	01010101
V	086	01010110
W	087	01010111
X	088	01011000
Y	089	01011001
Z	090	01011010

PUBLIC KEY CRYPTOGRAPHY

In public key cryptography messages are sent in such a way that only the receiver can decode them even when the encryption method is discovered by someone who has intercepted the messages. The sender encodes the message using a very large number (n) sent by the receiver to establish a secure communication channel. What's special about n is that it is the product of two very large prime numbers, p and q, that only the receiver knows, ie n = pq.

This method relies on the fact that it is virtually impossible to find the factors of a large number if it has only very large prime factors. Even the fastest computer to date would take many thousands of years to find the p and q.

A simple role play

- A 3-digit prime number is assigned to each student in the class (see list below)

	263		241		409		311
	373		227		257		449
	379		313		347		479
	353		389		281		509
	397		349		307		331
	283		457		421		439
	463		251		317		503
	499		467		461		233
	443		359		293		
	491		401		223		229

- Your teacher will send out a large number (n) publicly to everyone.
- If you are the receiver of the message, dividing n by your 3-digit prime number will return an integer prime number.
- If you are not the receiver, you will need to try all other numbers until you find the recipient of the message.

Who was the receiver?

The idea is that Bob (the receiver) chooses two (very large) prime numbers, p and q, and then writes n = pq. Then n is used to code the message, but p and q are needed to decode the message. The clever bit is that only Bob knows p and q, though n is sent out publicly, because deriving p and q from just knowing n is virtually impossible with ordinary computers.

The problem is that quantum computers will be able to find p and q from any n shared publicly in a matter of minutes, leaving any communication that currently uses this method completely exposed to hacking.

QUANTUM KEY DISTRIBUTION

The idea behind quantum key distribution is to be able to generate truly random keys that are shared only between the sender and receiver. There are different ways to do this. One system, not yet widely used, uses quantum entanglement. Entangled photons or particles are generated and their spin is measured independently by both the sender and receiver. A. If the measurement of the spin of each particle is measured along the same axis, the receiver (Bob) knows that the sender (Alice) will have measured a spin opposite to the spin measured by Bob. B. If the spin is measured by Alice and Bob along different axes, the spins will be uncorrelated and those measurements are discarded by both. C. To generate a secure key Bob and Alice share publicly which orientations (axes) they used to measure each spin, but they do not share their results, because Bob knows that for all the spins measured along the same axes by both, Alice will have measured a spin opposite to Bob's. D. To check that nobody is trying to steal their secret key, Alice and Bob publicly share a few spin measurements obtained through the same axis. If they record a number of errors, they know that someone has been trying to hack into their communications, because the hacker cannot know in advance the orientation used by Alice and Bob in their measurements. If they tried to be a 'man-in-the-middle' it is very, very likely they would emit a particle with the wrong spin which Bob would disregard.

Try this simulation https://goo.gl/X4xXTF and answer the questions below. Make sure you read the 'Introduction' first and look at the 'Step-by-step explanation' to gain better understanding of the process. Also, these two videos might help you understand quantum entanglement and quantum key distribution better: https://goo.gl/fHWPMf and https://goo.gl/Zp4o4r

1 Why does Bob need to invert the values of the outcomes measured along the same orientations as Alice to produce his key?

2 Activate all display controls, set the orientation of the SGA devices to 'Random orientations' and allow Eve to intercept and resend particles by clicking on the button 'Eavesdrop!'. Click on 'Fast forward 100 particle pairs' and click on 'Let Alice & Bob compare 20 bits for errors'. Make a note of the number of errors found and repeat a few times. What did you notice? Can you explain why this happens?

3 Why isn't Eve able to resend a particle to Bob always with the correct spin?

4 Why is it important that Alice and Bob measure each particle independently of each other and choose random orientations?

5 Explain why quantum entanglement ensures the key is shared securely between Alice and Bob.