Shout it out!

With the Vignère Cipher we saw how ciphers can be made to work if the sender and receiver have shared a secret key. But how can they share the secret key in the first place if there is a risk that the communication is being intercepted?

Today, instead of cracking a code, we are going to take part in a three player role-playing game. We're going to call the three roles "Alice", "Bob" and "Eve". Decide who you want to be.

- Alice and Bob are members of the Rebel Alliance, but in bases on different planets. They communicate by radio signal.
- Eve is part of the evil Empire and is intercepting radio signals. (I will be Eve, because I Am Your Father.)

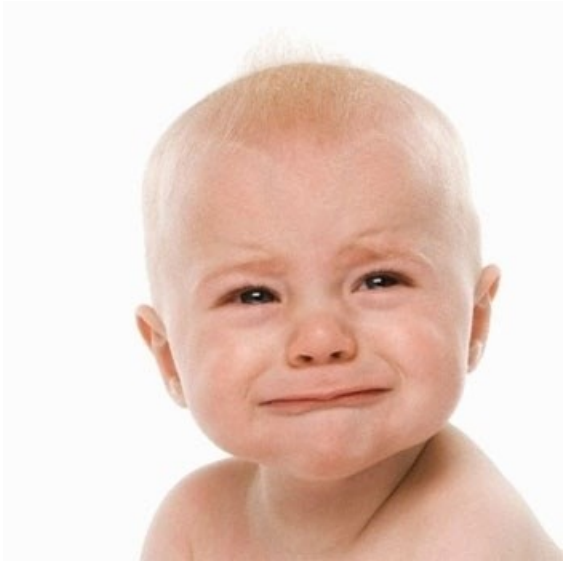
Bob needs to tell Alice the value of a secret key, which will allow Alice and Bob to securely encrypt and transmit rebel plans. If Eve finds out the secret key, the rebel plans will be discovered and the Empire will win the war.

To role-play the game there are a few rules:

1. All three players are in the same room.
2. Bob will think of a secret key between 1 and 100.
3. Bob must somehow tell Alice what the secret key is, without Eve finding out.
4. The only communication method allowed is **shouting**. No whispering, no hidden messages, no hand signals, no telepathy, no using of the force, no cheating.
5. To beat Eve, Bob and Alice must both write down the same number without Eve finding out what it is.

Let's Go!

Stuck already? Sad face.



Let's start again using these instructions

Step 0: Bob thinks of the secret key.

Bob thinks of the secret key. The first time you play the game, set the number to be less than 100 to make the maths easier. Call this number M. Bob secretly writes down the value of M.

We'll now use an approach called "**asymmetric cryptography**" to securely send the the value of M from Bob to Alice. We will do this using a "public key" and a "private key" pair. These keys have the strange property that a message encrypted with the public key can only be decrypted by someone who has the private key.

This is different to regular "symmetric cryptography" where the same key value is used to both encode and decode a message and you just reverse the procedure to decrypt (substitution ciphers, etc.).

Step 1: Alice creates a public and private key pair

Alice thinks of two large random prime numbers. The first time we do this perhaps choose values less than 1000 to keep the maths simple, but if you want to make life harder for Eve pick numbers larger than 1 billion. For secure on-line banking the prime numbers used need to be hundreds of digits long. There's a list of small prime numbers here https://oeis.org/A000040/list. Call these numbers P and Q.

Alice calculates P x Q, and calls the result N. Alice should write down the value of N and shout it out to Bob who also writes it down.

(If the value of N is smaller than M, Bob should let Alice know. If this happens, pick some larger prime numbers P and Q and try again.)

Alice calculates (P-1) x (Q-1) and calls the result Z.

Alice picks a random value between 1 and Z. Call the number E. Make sure that E and Z don't have any common factors; a prime number would work. Write down the value E and shout it out to Bob who should write it down. E is the "public key".

Alice must now find a value D, so that E x D divided by Z has a remainder of 1. The maths here is a bit complicated, but a special calculator can do it easily for you!

A calculator which can help find D: https://www.dcode.fr/modular-inverse

- *Type the value of E into "Integer Number A" field*
- *Type the value of Z into "Modulo N" field*
- *The result, on the left side of the page, is D*

Alice writes down the value D**.** She much not let Bob or Eve know this value. D is the "private key" so it must be kept private!

Now Alice, Bob and Eve all know the value of N and E. Only Alice knows the value D.
We are ready for Bob to encode and transmit the value M!

Step 2: Bob encodes and transmits the secret key

Bob calculates M to the power E modulo N, and calls the result C, and shouts out the value of C. C is the encrypted secret key.

A calculator which can help Bob calculate C: https://www.dcode.fr/modular-exponentiation

- *Type the value of M into the "Number A (Base)" field*
- *Type the value of E into the "Number B (Exponent)" field*
- *Type the value of N into the "Modulo N" field*
- *The result, on the left hand side of the page, is C*

Step 3: Alice decodes the secret key

Alice calculates C to the power of D modulo N. This value should be M, the secret key! We have managed to get this information from Bob to Alice without Eve finding out. Alice writes it down.

A calculator which can help Alice calculate M: https://www.dcode.fr/modular-exponentiation

- *Type the value of C into the "Number A (Base)" field*
- *Type the value of D into the "Number B (Exponent)" field*
- *Type the value of N into the "Modulo N" field*
- *The result, on the left hand side of the page, is M*

The Rebels Win!

If they haven't made any mistakes, Alice and Bob should now be able to demonstrate that they both have the same value M written down. They have managed to securely exchange a secret key which can be used to encrypt and transmit the rebel plans. The Empire cannot decode the plans, and the rebels eventually win the war.

More Information

This is the approach used by web browsers to talk to web sites using HTTPS. When you first use a web site, your browser and the web server will use asymmetric cryptography to exchange a secret session key, so that any network snoopers cannot see the key. The session key can then be used safely for encrypting the rest of the data while you use the web site (web pages, uploaded bank details, etc.). The session key can be thrown away when you close your browser.

The maths behind asymmetric cryptography relies on the fact that it is extremely difficult for Eve to calculate D, the private key required for decoding the message, based on N and E which are public information.

In a real system the prime numbers P and Q would be hundreds of digits long, to make it too time consuming for an attacker like Eve to try all the possibilities to find D. This problem has "*exponential*" difficulty; each time Alice adds a single bit to the sizes of P and Q, this roughly doubles the amount of work Eve needs to do to crack the encryption. Small prime numbers make it easy to crack; big prime numbers should make it nearly impossible.

If you're interested in more information, Google "*asymmetric cryptography*" or "*RSA algorithm*".