Activity Guide - Recap: The Public Key Cryptography Widget

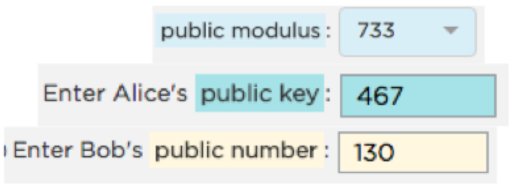
This document assumes you have used the Public Key Crypto Widget.

What does Eve have to do to crack the message?

It turns out that figuring out what numbers to plug in to crack Alice or Bob's secret numbers so that the equations work out is essentially random guessing. If you want to read more about why check out the "How and why does it work?" document which gives a deeper explanation of the math behind the widget.

What does Eve know?

Eve knows only public information announced by Alice or Bob

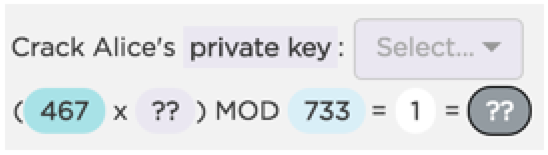
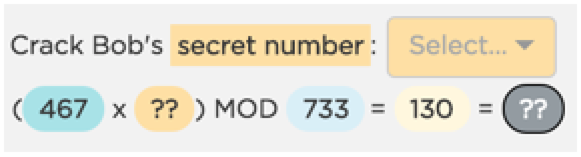


public modulus : 733

Enter Alice's public key : 467

Enter Bob's public number : 130

To crack the message she must use this info to guess Alice's Private Key or Bob's Private number.

To Crack Alice's Private Key...	...or to Crack Bob's Private Number
 Crack Alice's private key : Select... (467 x ??) MOD 733 = 1 = ??	 Crack Bob's secret number : Select... (467 x ??) MOD 733 = 130 = ??
To crack Alice's private key Eve must figure out what number to multiply by 467 then MOD by 733 to come out = 1 (why 1? It's explained in the *How and Why does it work?* document)	To crack Bob's secret number Eve must figure out what number to multiply by 467 then MOD by 733 to come out = 130.

Recap - Properties of Public Key Encryption:

- Alice and Bob did not have to agree on anything, or communicate ahead of time.
- Alice and Bob only exchanged information in public, right in front of Eve. Eve could even chose the public modulus if you wanted to!
- Eve would have to guess either Alice's private key or the secret number Bob is trying to send.
- *Note:* The encrypted messages only go one way. If Alice wanted to send a message to Bob, Bob would have to generate his own public/private key pair.
- However, because the message can only go one way, the sender can feel safe that ONLY the intended recipient can decrypt the message.
- The *real* public key crypto does not quite work this way. What we have setup here emulates many important properties, but it is in reality not hard for a computer to crack - just hard for you.
- The *real* version is called RSA encryption and rather than simple multiplication it uses exponentiation in combination with properties of modulo to create even larger numbers that are even harder to guess.

What do you actually need to know?

- **Cryptography has a mathematical foundation.**
- It relies on **asymmetric** keys, which you can make using numbers and math.
- The **modulo** operation acts as a one-way function.
- When you multiply big numbers and mod them by other big numbers, it's really hard to figure out what the original numbers were; the technique is essentially reduced to random guessing.
- The security of publicly known encryption protocols is based on the fact that cracking a message by brute force would take an unreasonable amount of time.
- With a sufficiently large modulus (say, 256 bits, which would be roughly a 77-digit number), random guessing would take an unreasonable amount of time. Even if you had millions of computers working on it constantly, it would take trillions of years.
- Because the method of encryption is public, it actually *increases* the security, because good guys and bad guys know how hard it is to crack.